

QL Today

Volume 3
Issue 5
Jan./February
1999

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

**Happy
New Year!
The Euro
is here!**

Q40

?

**SMSQ
/E**

Lin

?

**QDOS
Classic**

Contents

3	Editorial	
4	News	
7	The right EPSON printer - a perfect add-on for the QL!	R.S.R.Thain
8	Kan U Spel Uzing Yore QL?	Geoff Wicks
10	What are all these Handshake Lines? - Part 2	Nasta
12	The Nine Trials of TF Services	Tony Firshman
14	Gee Graphics! (On the QL?) - Part 8	Herb Schaaf
20	New ED Drives	Mark Knight
22	Assembly Language Programming - Part 3	Norman Dunbar
28	QDOS Bugs - Part 3	Mark Knight
31	Letter-Box	
33	Attachments?	Ian Pizer
34	The future-proof QL clock	Mark Knight
36	UNIX-ifying QDOS	Tim Swenson
40	Hard Disk Utilities	Dilwyn Jones
42	Paragraph v1.01 - a first look	PBailey
44	QD98 - Review	Dave Westbury
51	K-BASE - an easy to use Database	Darren D. Branagh
53	Byts of Wood	Roy Wood

Advertisers

in alphabetical order

Jochen Merz Software	13, 33
PROGS - Van der Auwera	23
Q-Celt Software	19
QBOX USA	11
QBranch	38, 39
Quanta	49
RWAP Software	29
W.N. Richardson (EEC)	47
TF Services	43



QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
Im stillen Winkel 12 Fax +49 203 502012
47169 Duisburg Box1 +49 203 502013
Germany Box2 +49 203 502014
email: JMerz@j-m-s.com

English office:

Q Branch Tel. +44 1903 200212
The Bank Volt Mobile +44 836 745501
6 Coronation Buildings Fax +44 1903 208070
Ham Road Fax +44 1273 381577
Sussex BN11 2NN email: qbranch@qbranch.demon.co.uk
United Kingdom

Editor:

Dilwyn Jones Tel. +44 1248 354023
41 Bro Emrys Fax +44 1248 354023
Tal-Y-Bont, Bangor email: dilwyn.jones@dj.softnet.co.uk
Gwynedd
United Kingdom LL57 3YT

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque. Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 15 April	Issue 2: 15 June
Issue 3: 15 August	Issue 4: 15 October
Issue 5: 15 December	Issue 6: 15 February

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine is copyrighted and all material published remains the property of **QL Today** unless otherwise specified. Written permission from **QL Today** is required before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

A happy QLing new year to you all! And especially to our merry (not too merry I hope!) band of regular contributors, thank you all for your hard work for QL Today, where would I be without you? And thanks to Jochen Merz too - although my name appears in lights in this magazine, Jochen is the hardest working of the pair of us, I hope readers appreciate his hard work in laying out the magazine and getting it printed.

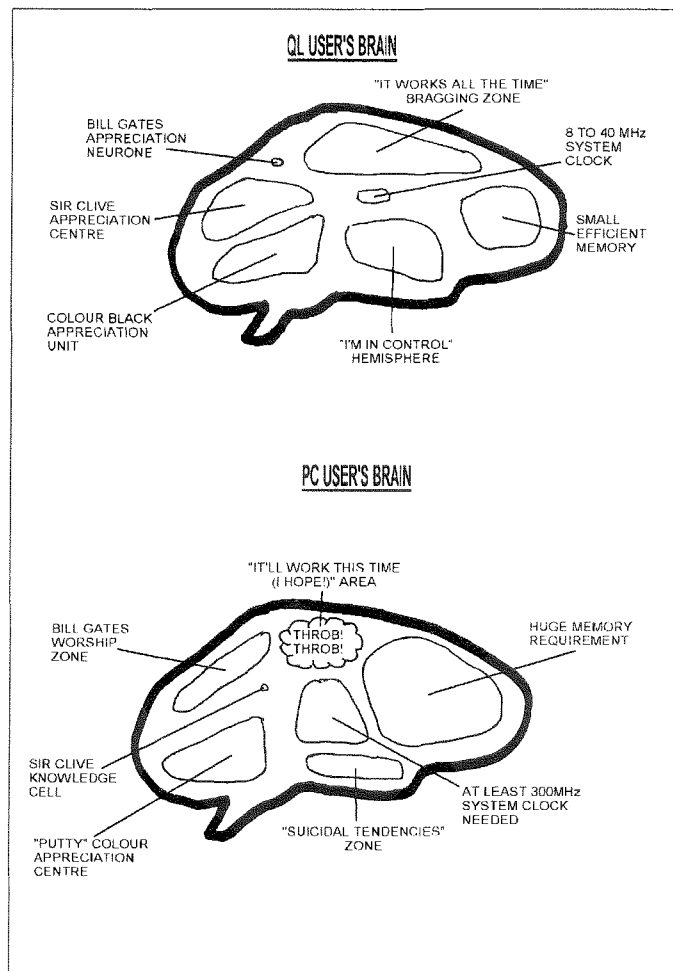
Looks like we'll have a pretty exciting year ahead. Negotiations on the sale of the Q40 have been concluded, it has an operating system, now all it needs is customers, and I'm sure there'll be plenty, especially when people learn that Tony Tebby is using one as one of his main computers!

I got a MinisQL for Christmas (just!) and after the problems I had with the tower cased Aurora before it, I'm glad to say that the same Aurora board works well in my MinisQL system, so full marks to Aurora, while everything else around it was on the blink, it stayed in one piece and worked better than ever in the final system. Paragraph, the new pointer driven wordprocessor, is out (report in this issue), a new version of QemuLator and a trial version of QPC2 are out, and the colour drivers (in the guise of SMSQ/E 3) should be with us before not too long, rumours continue of work being done in Austria on the basic software required for internet access, the new Turbo Toolkit is out (see below) and plenty of other things are happening too!

OOPS - Two significant errors got through the net in the last issue, we got the date of the Croatian meeting wrong (it should have been December) and I also got the name of Christopher Cave's file viewer program wrong - I called it MText whereas its real name is MView. Apologies to all concerned.

The next issue will, I hope, see another cover disk with the magazine. Let's hope I make a better job of that! One of the things we hope to include on it is the new freeware release of Turbo Toolkit thanks to the hard work of Mark Knight and David Gilham.

And speaking of the next issue, I'd love to receive a few more articles from you the readers for publication. If you've used a nice little program or new piece of hardware, write and let the other readers know about it. Equally, if you'd like to offer comment via the letters page, or pass on some hints and tips or short, useful listings, we'd love to hear from you. We are a little short of material for the next issue as I write so please do write in with your contribution on floppy disk or by email, the readers would love to hear from you!



Cartoon

NEWS

RWAP Software

(Rich Mellor)

I have now been able to release v1.5 of Deathstrike which is yet another game formerly sold by Talent Software. This is a clone of the classic arcade game Scramble, in which you control a space ship as it travels through an alien base, destroying as much equipment as possible in the search for the mothership. This is a colourful and fast version which will now work on all versions of the QL (although I have noticed that the game does not seem to work with a SuperHermes keyboard for some reason). I can now also supply a wide range of IBM Compatible PCs and components (including monitors, mice and printers) at competitive prices, and should be able to ship them with QPC ready installed if required.

PROGS

Joachim Van der Auwera sent us an email suggesting we point out that ProWesS users who have Web access can get an update from his Web site at <http://www.triathlon98.com/PROGS/>

This may prove useful information for ProWesS users who are trying to get early versions of the new word processor Paragraph working, for example, as that reportedly requires up to date versions of ProWesS to work correctly.

QemuLator V1.1

A new release of Q-emuLator for Windows (version 1.1) is available at the Q-emuLator's web site:

<http://www.geocities.com/SiliconValley/Heights/1296/q-emuLator.html>

The two most relevant new features are support for sub-directories in the file system and read-only access to QXL.WIN files. Graphics speed has been improved, the maximum QL RAM amount has been increased to 16 MB, and for the first time a QL ROM is included in the Q-emuLator package, though it doesn't work with more than 768K of QL RAM. (It is a Minerva 1.89 ROM image, now public domain).

Another interesting feature is that temporary registration codes are now freely available: if you are not yet a Q-emuLator registered user, but you want to try the unlimited version of

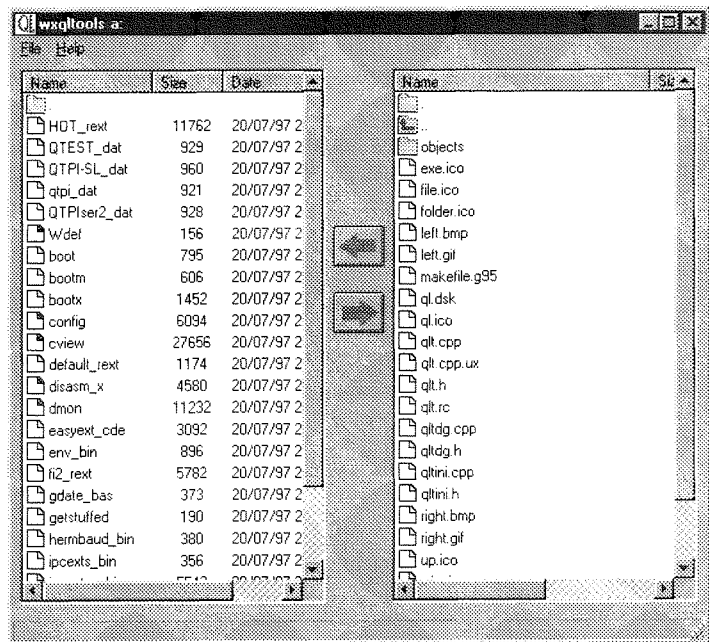
Q-emuLator, just ask to the author by email (the address is danielet@geocities.com) and he will send to you a free temporary registration code that will last for about two weeks. Be sure to include your full name in your request. Work has already started on Q-emuLator version 2, that will have a faster and full screen QL display.

Jonathan Hudson

In the last two months the following ****free**** packages have been uploaded to www.jrhudson.demon.co.uk
perl4qdos: Port of perl to QDOS. perl 4 rather than perl 5,

alas. Package contains binary, documentation, source code and many examples. Examples are mainly QDOS specific and the port adds support for many specific QDOS features, including the graphics traps.

A comprehensive (10 page) article on perl4qdos will be published in Quanta, January 1999. The article will also be available as HTML from www.jrhudson.demon.co.uk in Jan 99 for non-Quanta members.



wxqltools GUI package under Windows (95,98,NT) or Unix to read and write QDOS floppies under foreign operating systems. This program is a graphical front end to qltools.

gdbm173: The GNU database manager. GNU dbm is a library of routines that manages data files that contain key/data pairs.

The access provided is that of storing, retrieval, and deletion by key and a non-sorted traversal of all keys. Source, documentation, c68 library, examples in perl and 'C'.

nroff: nroff port for QDOS. Converts Unix man pages and other nroff format files to text. Includes man and ms macro packages.

catdoc: Converts catdoc converts MS Word documents (inc Word 97) to text. Archive contains executable, source and documentation. Ported from unix catdoc program of Victor Wagner.

unzip540xQ: unzip 5.40. Binary and documentation. Source code available from **ftp.cdrom.com**. The QDOS version is provided (experimentally) as a self extracting object file.

LRESPR unzip540xQ.obj
And follow the on screen instructions.

Dutch Thesaurus now available

JUST WORDS! has released a Dutch language thesaurus. The program is pointer driven and is based on the popular English QL-THESAURUS. It has a database of 13,500 words and phrases.

NL-THESAURUS costs fl. 20,- or £7. Payment can be made in guilders by direct transfer to Netherlands Postbank number 4111942 (G.T. Wicks) or by Eurocheque in pounds.

JUST WORDS! is preparing a series of freeware dutch language dictionaries in QTP and SPELLCHECKER formats conforming to the latest spelling rules. Dictionaries of 52,000 and 70,000 words are currently available, and a larger dictionary should be available in the new year.

Geoff Wicks, 28 Ravensdale, Basildon, Essex, United Kingdom. Telephone from Belgium and the Netherlands 00 44 1268 281826

Turbo Toolkit V3b27

Mark Knight and David Gilham have now released version 3b27 of the Turbo Toolkit as freeware. QL Today hopes to distribute a complete copy on our next cover disk and it

should also be available from the usual sources of QL freeware. Below you will find the major list of changes for this version.

This version has been produced by disassembling the last official version (v3.22). Interestingly, only 7 significant bugs were found and contrary to popular opinion, no direct system variables access etc were found.

It has been tested with QL ROM versions from AH up to SMSQ/E (on a variety of hardware platforms). Mark says that some Turbo compiled programs which failed on modern systems may now work with this new toolkit.

Keywords updated include TYPE_IN, ALLOCATION and DEALLOCATE, COMMAND_LINE, SET_PRIORITY, COMPILED. The latter now even works with QLiberator.

New keywords added include LONGINTEGER (converts a 4 byte string holding an integer value to a long word) and LONGINTEGER\$ (converts a long word into a matching 4 byte string), both useful for when you need to store 32 bit long words in strings or directly in memory. POKE_F pokes a 6 byte representation of a floating point number into memory. FWINDOW% helps with handling hi-resolution screen windows - it is a function which returns an error code number making error handling easier.

The file of demonstration basic program listings (DEMOS_bas) has also been slightly updated to use the SYS_VARS function to access system variables rather than hard coded addresses, for example.

The distribution includes a program to patch startup code in Turbo compiled programs, to try to make some older programs work on modern platforms. It contains code written

by Davide Santachiara to patch among other things two hard coded system variable accesses in the startup code for older Turbo compiled programs.

Turbo Toolkit can now be used with QLiberator. In addition, it is possible to link it as an SMSQ/E module for those who know about such things. Since the toolkit is freeware, the old runtime version (RUNTIME_EXTS) is no longer required - you can simply include a free copy of the toolkit itself with software sold or given away!

QBranch News

The Q40 should be available from the end of January - middle of February. The product will cost 350.00 UKP and this will include 16Mb Ram (upgradable to 32Mb) and the I/O card. All the end user will need is a case, keyboard, floppy /hard disk, monitor and mouse. No QL chips or other hardware is required. The system will be shipped with the QDOS classic system on board. SMSQ/E for the Q40 is nearing completion and will cost extra although there will be discounts for existing SMSQ/E users. The price may be reduced if we can obtain lower priced 68040 chips than those currently quoted.

TF Services are currently looking into other boards to plug into the ISA slots and an ISA slot extender.

Masterbase, the pointer driven front end for DBAS should be available for release soon. This will be supplied on two floppy disks with one containing the latest DBAS Public domain release.

Possible visitors to the Hove show include Zeljko Nastasic and Peter and Klaus Graf. Jochen Merz will definitely attend. Masterbase, Q40 and QD 98 will be demonstrated there.

Extra: We are asking people to place firm orders for the Q40 because this is an expensive product and we would like to have at least part of the money guaranteed before we embark upon it. The first 25 people to send us Cheques or credit card details will receive free system upgrades and a reduced price on SMSQ/E when it is released. No cheques will be cashed or Credit cards debited until we are ready to ship the first units but just having the pledge of cash will make us feel more secure. If the final price is lower than that quoted then we will refund the difference.

Z88 News

Alchemist Research have sent us a copy of a CD-ROM called Z88 Heaven which contains over 1,000 Z88 programs and Z88 emulators to run on the PC. As many QL users also use a Z88, we decided this may well be of interest to our readers. We would also be interested to hear from anyone willing to review this CD-ROM for us.



The CD-ROM contains the entire collection of the Z88 User Group Software Library, and files from the United States Z88 User Group library. There are Eprom image files and ROM images which can be transferred to a

Z88 and then programmed into an EPROM, the most prominent of which is the Z88 Forever Collection, full of the latest Z88 applications. It also includes Windows 95 utilities, allowing you to "transfer files, read BASIC and Pipedream files directly." Finally, the CD-ROM contains the Z88 Source Book and several very useful articles and guides detailing every aspect of the Z88.

The CD will be available from Bill Richardson at a price of £15 or £12 to Z88 User subscribers.

Z88 with internal memory expansion

W. N. Richardson & Co now offer internally expanded memory Z88s with either 128k or 512k ram, with OZ4 operating system version. This memory replaces the internal 32k, consumes about the same working and standby current, and can be directly used by Basic and Pipedream as workspace as well as for normal file storage, though the Z88 reserves a little for internal use. An upgrade service for your existing machine is also offered. See Bill's adverts for prices.

JMS

Quite a lot of news this time from Jochen Merz Software. QD98 is released (in fact, it was released in Croatia beginning of Dec. 1998 - so the name is still OK). A detailed description of its new features plus some screenshots (done with QPC 2!) can be found in the JMS ad on page 13.

By the time you read this, SMSQ/E 2.91 will be released. It

has the Euro-Symbol on character code 181, and it will translate them to character 164 (this is the Euro Symbol in the character set Latin 0). I think we will have a dedicated Euro-article in the next issue. (Dietrich Buder has done quite a lot of work on the Euro, created Download-fonts etc., so we can base it on his experience). If you want to use the Euro, then I suggest that you get the new SMSQ/E first. Anybody out there who has created a download-Euro for non-EPSON printers (HP Deskjet or Laserjet etc.)? Please get in contact with QL Today so that we can offer a solution for a variety of printers.

Lots of news on the JMS website:

<http://www.j-m-s.com/smsq/>

You can see information about the new QD98 if you want to see it in colour.

Follow the link to Marcel Kilgus homepage: you can download an alpha version of QPC which works and supports most features but for serial port, parallel port and writing to flp and win (but it's a demo, so it will never do this anyway in the demo!). Why not give it a try, you can now switch between QPC 2 and other Windows applications without having to reset the machine - similar to the way you switch between jobs in QDOS. Have a go, if you encounter problems, tell Marcel, otherwise enjoy it!

The QL News List is working again. I am using a different provider who double-checks that you want to get in the list by sending a reply email to your subscription which you have to return - email addresses have to match, of course! Subscribe via my website. If you have email access only, then send an email to "add2qlnews@j-m-s.com" and I will add you manually. If anybody needs bulk DD disks, please contact me, I've got masses.



The right EPSON printer - a perfect add-on for the QL!

R.S.R.Thain

It comes to us all sooner or later. In January 1998 my Olivetti 24-pin dot-matrix colour printer began giving trouble. First of all, my usual suppliers told me that they were no longer stocking colour ribbons for it - insufficient demand. After looking around I got black OEM ribbons. Maybe the printer resented this indignity, for a month or so later it became temperamental, refusing to move to the start position, or to let me examine or reset the defaults. After some struggling I decided to retire it. Half-remembering dire warnings in QL TODAY and QUANTA magazine about modern printers vis-a-vis QLs I searched my old copies for guidance. It was not altogether successful, but I did come up with a statement by Jochen Merz that the Epson 600 and 800 series should be all right.

So I went to the local computer supermarket which as usual was having a Sale, and looked at their printers (20% off). It was a bit disconcerting to find that the Epson 800 had been replaced by the 850. Had it been "dumbed-down"? After a talk with the assistant (helpful, but I don't think the problem really registered) I decided "probably not", and returned home £279 poorer with a very large cardboard box. **[Editor: as far as I know, the main difference of the 850 is that it is even faster than the 800 - and I am very happy with the high speed of the 800!]**

I use a QL with Super Gold Card (upgraded by Qubbesoft in the middle of all this to an

Aurora), and a QXL2 in a 386DS PC, both with SMSQ/E. The old printer was operating in LQ2550 mode. My communications with it were with Printer-master (an excellent program for ad hoc control), or with the Xchange drivers, or in some of my programs by commands issued directly from SBASIC. I also used SDUMP quite a lot (printing movement cards on 160 gsm paper for a local Bridge Club), but the Olivetti had trouble with this and I kept my original printer - a 9-pin Shinwa clone. GPRINT-prt was unsuccessful also, so that I could never make a print from EASEL.

I have just finished testing all these with the new printer. To my delight they functioned perfectly without modifications to programs or even to the parameters of SDP_SET!

GPRINT_prt makes tasteful, if somewhat pale, grey-shaded prints of my graphs, filling the whole width of the paper. **[Use Jochen's filter24-program (available free on his BBS) which converts greyish 9-pin dump output to perfect 24-pin output!]** Incidentally, I have tried using SDUMP to get colour prints from EASEL. It works, but you also get EASEL's borders, the buttons, and whatever else happens to be on the 640x320 screen. I am indebted to the splendid SBASIC/Superbasic Manual by Rich Mellor and others for the information (which I'm sure I ought to have known) that SDUMP can be used to print a specific part of the screen, but I have to do that from basic, and as soon as

you enter basic, eg from a Hotkey, EASEL vanishes! **[Set it on a HOTKEY - we'll show you in the next issue how it works!]** I tried reducing the screen, but, as the Manual says, SDUMP stays with the original screen size, and you get items in the dump which were invisible on the screen! So that is something I have given up for the time being. As a final test, I tried downloading the QLQ fonts (another thing that the Olivetti did not do very well). The results were excellent.

As one might expect, the print quality is better than that of the 24 pin printer, which itself was pretty good. It seems strange to be using cut sheets instead of fan-fold paper, and even stranger that instead of being able to read every line as it is printed, the paper disappears into the printer and emerges after a while with the print upside-down. When switched on, the printer sometimes makes mysterious noises for a time before giving the go-ahead, but that can be borne.

The printer Manual inevitably devotes most of its generous space to PC's and Macs. It does, unlike a 600-series Manual I looked at, give a list of all the control codes that the 850 will accept; but doesn't say what they do! Provided you are content with dot-matrix capability and are familiar with the old Epson codes it can be used without further documentation; but the 850 can of course do much more - extra fonts, some of them scaleable, very precise positioning of the printing, and several esoteric routines for high resolution graphics. And so I bought the ESC/P2 Reference Manual for £43.26 including next-day delivery from Unicomp Ltd (01732 780303).

This weighty volume (nearly 3.5lbs), dates from 1994, and therefore has almost no references to ink-jet printers, except in parts, mainly graphics, which have been revised since. But it is no less useful for that. In addition to an exhaustive and illuminating description of all the control codes used by Epson printers since the beginning, it includes a series of essays on subjects such as "Set the printing area" which I read with great fascination and partial understanding.

I'll try to implement some of this as time permits, but for now that's about it. The printer weighs 14.3 lbs, has a footprint (when shut down) of 18.7x10.7in. (475x274 mm). The paper support adds about 5in. to the depth. Cartridges cost about £16 (black) and £18 (colour). According to the Manual, at 360 DPI a black cartridge will print 900 A4 sheets, and colour 300. Prices are for Epson cartridges: "compatible" ones seem to cost between £7 and £11, but I'll stay with Epson while the machine is under warranty.

Editor: I'm glad to read that some people follow the advice and are very satisfied with the result. You will find more details about the very recent, cheaper model Stylus Color 740 in the "Letter Box" section of this magazine.

EPSON Germany can provide special add-on pages (3 sets) for the three generations of EPSON ink colour printers - you should be able to get them from EPSON UK too if you bought the ESC/P2 Reference Manual. Provided, enough interest exists, I can dig out the old filter24 and explain it in the next issue. It takes 9pin-EPSON output and prints it on 24pin printers in solid colours, unstretched.

Kan U Spel Uzing Yore QL?

Geoff Wicks

The last issue of QL-Today contained a review of my free-ware program "Spelling-Crib". Most people tell me I should have added it to my program range and charged a fiver, but I preferred to make it public domain. The program was quick and simple to write, mainly because it uses routines from existing programs, and I wanted to use it to encourage discussion on QL spelling aids and spellcheckers. I also had the less altruistic motive of reaching QL users who have not bought my programs. Every time they use Spelling-Crib the JUST WORDS! logo and distinctive house style will be on their screen as a free advertisement for my software. If Adidas and Nike can do it, why can't I?

Spelling-Crib was conceived at a workshop when a customer suggested the use of SOLVIT-PLUS as a help for the occasional difficult word. He had most of the spelling aids available to QL users, but found these of limited help. I felt a less cumbersome method than he suggested would be to combine a few short routines from SOLVIT-PLUS with other routines developed for QL-THE-SAURUS, and then use the stuffer buffer to link with a word processor or text editor. The program was written and tested in about 2 months, although, as Dilwyn Jones discovered, it still has a few minor bugs. At the moment it uses a memory hungry plain text file, although I hope to write a version that will use a QTYP dictionary.

Many QL users are not happy with the current spelling checkers and aids, but few have

concrete ideas of what a spell checker should do. As I have written before in QL Today, people have strong personal preferences over spell checking. Some want to check as they type, while others want to check on completion of a document. Some want suggestions for alternative words and others do not. Some want as large a dictionary as possible, but others prefer a small one.

The QL has three commercial and one public domain spell checkers. Each has its strengths and weaknesses, but it is only the public domain one that can be described as "intelligent" in its suggested alternatives for a misspelt word.

I know nothing about the oldest spell checker, "SPELL-BOUND", which is rarely used these days. The other two commercial spell checkers are "SPELLCHECKER" and "QTYP". SPELLCHECKER is the dedicated checker for the word processor Perfection, and although it is possible to use it to check text files, it is of limited use outside of Perfection. It does not provide suggestions for a misspelt word, and also has some restrictions that limit its usefulness for many non-English languages. Nevertheless there are many Perfection users who are happy with this level of spell checking.

In contrast QTYP is a generic spell checker that will work with most QL text programs. It is not strictly a spell checker, but a resident extension that enables you to incorporate spell checking in your own programs. Text87 plus 4 users can use QTYP in two ways. They can either use it in its native state for check as you

type spell checking, or on (part) completion of a document using Text87's own implementation of QTYP spell checking.

QTYP works by comparing each letter of a word against the list of words in its dictionary. When a mismatch is found, a warning sign goes up. If you wish, you can examine a list of words for the correct spelling. To access a QTYP dictionary you need to know the first letter or letters of a word. Spell "accommodate" as "accomodate" and you will quickly find the correct spelling using QTYP, because it is looking for the root "accom". Spell "surprise" as "suprise" and you will find it hard to discover the correct spelling, as QTYP is looking for a root of "supr". Spell "cinnamon" as "sinnamon" and QTYP will be of no help whatsoever.

Most PC word processors have an intelligent spell checker, which will suggest alternatives for a misspelt word including mistakes like "sinnamon". When Spellchecker and QTYP were first released, the QL was still using a slow processor and had a maximum memory of under 1MB, and it would not have been possible to have had an intelligent QL spell checker. They use lengthy, complicated and specially constructed dictionaries containing lists of homophones, or words which sound the same, but are spelt differently, such as "sauce" and "source".

The QL now has the memory and speed for an intelligent spell checker using similar programming to those on the PC, but in practice it is unlikely that one will ever be produced. It would require a programmer with both the time and the specific language knowledge to construct the dictionary, and I

doubt if such a person exists. There is a further complication. Present QL spell checkers use simple dictionaries - basically they are just lists of words in compressed form - and it is possible for any QL user, even if he has no programming skills, to write his own dictionaries. Both the Italian and Dutch QTYP dictionaries were written in this way. The dictionary for an intelligent spell checker would have to be separately constructed for each language, again by a programmer with the necessary language knowledge and time for the task.

But why should we copy PC programming? If we are just going to ape what happens in the PC world, then we should be using PCs. The history of the QL is full of innovative and creative thinking to solve what many once thought were insolvable problems. And this has also happened in spell checking. There is an "intelligent" spell checker for the QL that uses different techniques from the PC spell checkers.

This spell checker, Suggest by Lester Wareham, uses QTYP to do its spell checking, but in an unusual way. Basically the program assumes that spelling mistakes and typing errors have what you might call a logical cause. We hit the wrong keys on the key board; we type two characters instead of one or vice versa; we type words as they sound and not as they are spelt. If Suggest comes across a misspelt word, it looks through a set of rules, and uses these to devise theoretical words the writer could have intended to use. It then checks these words against the words in the QTYP dictionary, and if they are valid words, suggests them as alternatives for the misspelt word.

Unfortunately Suggest is not bug free. It will not run on all QL systems, including the early Sinclair ROMs and the Atari TT. It does not always replace a misspelt word accurately and sometimes gives it an extra first letter. Nevertheless I am impressed by the program, and especially the innovative and creative thinking behind it. It is a pity that it was not developed further.

In this article I have attempted to assess the present state of QL spell checking, and explain why QL spell checkers are limited in their scope. In my opinion an intelligent spell checker for the QL could not use traditional programming methods, but one could be written using the QTYP extensions on the lines of Lester Wareham's Suggest. It may even be possible to devise a set of "compromise rules" that would apply to most languages, and, if necessary, supplement these with a short set of rules for each specific language. There is, of course, an important proviso. In my opinion Lester Wareham has a right to an intellectual copyright for his idea, which should be respected.

I am not one of the elitists when it comes to spelling, and do not see poor spellers as being inferior or lazy. There can be many reasons for poor spelling, some of which are physical in origin. I would be interested in hearing what readers of QL Today would like to see in a spell checker, or of their ideas for other spelling aids. I would also be interested to hear from programmers who have ideas for spell checking that they would like to put it into practice.

■

What are all these Handshake Lines? - Part 2

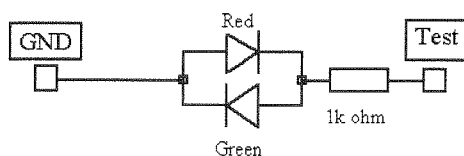
Nasta

Determining DTE and DCE, quick and dirty (but effective) solutions

As can be seen, correct cabling relies heavily on the correct identification of the device type - in fact, just about the only way it will work if you get the type wrong, is if you get both device types wrong and also if all the handshake signals are implemented. As can be seen from the size of the table above, there is much working against this case! The simplest way to determine what type the device is, is to look for output signals that should be active when the device is powered up. Here are some tips:

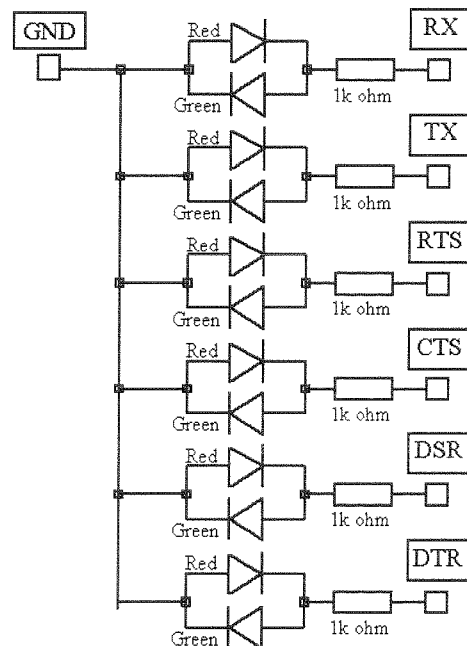
- Using a voltmeter to determine device type:
First of all, disconnect all cables to the port to be sure you are measuring signals only at this port, not the signals that might be coming from the other end of the cable. Take a 3k or thereabouts resistor (almost anything between 1 and 10k will do just fine) and connect it across the ends of a voltmeter (or multimeter set to measure voltage). Then, with this resistor connected, check the voltage between ground (GND) and the various lines. If you measure anything within legal RS 232 voltages (+3 to +15 or -3 to -15 Volts) you are looking at an output signal. Take note of the signal name, and use the DTE/DCE comparison table at the start of this article to determine what the device type is. If you find either RXD, DTR or RTS to be outputs, its a DTE, and if you find either TXD, DSR or CTS to be output, then its a DCE. Remember, it is enough to determine one of the three is an output to determine the type of the device correctly. In some cases the device in question may have to be told to 'open' the port in software or by an appropriate command (this will normally be the case with computers). The best candidates to start testing are of course RX and TX lines as these are mandatory. They will normally be 'inactive' if there is no data being sent from the device. There are exceptions, however, and the device described below can help with those.

- Building a simple RS 232 indicator:



The above diagram represents a dirt cheap and very simple RS 232C tester. The diodes are LED, i.e. light emitting. The device should be connected so that the GND terminal is tied to the serial port GND terminal, and the Test terminal is connected to each of the RS 232C signals in turn. If a signal is an output, the green LED will light up if its active and the red LED will light up if it is inactive. If you are testing RX or TX lines while data is being transmitted or received, the green LED will light up dimmer than for other output signals if the port is RS 232, and both LEDs will be lit dimly if it is RS 232C. The respective LEDs intensity will vary depending on what data is being sent. The device can be used instead of a voltmeter or multimeter exactly as described above. The device can be built inside the plastic parts of an old pen with a little care and attention to detail. It is possible to get two-color LED diodes that have two opposed LEDs in the same package, connected just as on the diagram above. Just note that if you use one of those, both LEDs lighting will produce an orange-ish light. The nice thing about the simple device above is that it does not depend on the type of connector used, unlike the device described below.

- A quick and dirty RS 232 breakout box:



By building a circuit not unlike a 6-fold tester described above, a simple RS 232 test box can be made. I will leave it to the prospective builder

to decide which one of the myriad RS 232 connectors that are in common use will be chosen for this project :-)

- Using an oscilloscope

When everything else fails, an oscilloscope will come in handy. However, I will not delve deeply into this subject as anyone that has a 'scope will probably already have ideas about the use of it to debug serial ports. Just bear in mind that the data on the RX and TX lines is actually inverted with respect to the actual bit values, so one should expect to see a picture not unlike a flipped over diagram given in this article, with start, data, parity and stop bits. It does take some tweaking to get it on the screen but it is possible. Needless to say there are even more specialized devices to debug serial ports (some costing on the order of a good car), which just goes to show that QL users are not the only ones having serial port problems.

Final word

You have probably noticed that I haven't even attempted to show all the different connectors

used for serial ports. This data can be found elsewhere for the port in question - manuals DO really contain useful data once in a while, you know! They make the number of possible variations even larger but not unmanageable - the key, as always is understanding what the signals on the ports do. With this knowledge most if not all ports can be wired together correctly. I hope I have managed to get at least some aspects of serial communication explained. This lengthy article however is not by far definitive word on the subject - in fact, whole books have been written about this. Still, I hope that anyone trying to connect two stubborn devices using serial ports, finds this article of use. Don't be discouraged if things don't work at first. Fortunately, serial ports are very robust and are made to be able to survive wrong connections - it seems like the creators of the standard knew that they are bound to happen. So, may your lines be connected properly, your baud rates chosen wisely, your parities the same on both ends of the link, and your bits flow freely :-)

■

WE SUPPORT SINCLAIR

QBOX USA

COMPUTER BULLETIN BOARD SERVICE

(810)254-9878

**We support *all* SINCLAIR COMPUTERS
(QL, SPECTRUM, ZX81, Z88, Thor, QXL)**

- Now in our 4th year on-line round the clock since October, 1993
- Full message area and File Download areas
- We carry all popular SINCLAIR message areas from Europe
- Calls from 14.4k--300 baud are welcome
- QBOX - USA runs on a SINCLAIR QL with Super Gold Card, Hermes, QUBIDE, 200MB drive, USR sportster 14.4 modem

→ NO FEES CALL US

The Nine Trials of TF Services

Tony Firshman

Hardware manufacture is a difficult business without the aid of God(s).

MPLANE sprang from the Qbranch stable and Keith Mitchell, who found a neat surplus notebook style case. A backplane was needed, but Qplane was too high.

I developed a low profile 2 x 2 backplane which was launched at the recent Byfleet workshop.

I produced a card board prototype, with motherboard (Aurora or original QL) and expansion (Gold Card, Super Gold Card etc) at the side in the same plane. Qubide sits above the expansion, leaving one spare slot above the QL motherboard, and even a half slot on Qubide. Now Keith found the cardboard prototype a little difficult to power up, but everything fitted - after minor (1mm) adjustments, even with the ROM slot fitted on the card.

..... so I committed the necessary £600 for board production.

The 4 layer expensive boards arrived 4 days before Byfleet. I built up one and tested the internally generated RS232 voltages and they were fine - gave the designed +10v/-10v.

Plugged in a QL motherboard - and DEAD. This is where the Gods start arriving.

(Gods 1) At that very instant, Roy rang up! "Hang on a minute" - I said, while I saw £600 melting away. Again at that instant I realised I had the components on the wrong side of the board, reversing all signals, including +5v/GND. This is unfortunately very easy to do, as PCB packages are not at all clear on this point - they call

sides 'TOP' and 'BOTTOM' My components, well most of them, needed to be on the bottom. While Roy was on the phone, I worked out that everything would be OK with components on the other side, but Keith had designed around the wrong cardboard version!

(Gods 2) Keith used the reversed production board (Motherboard and SGC were reversed) and it fitted and worked

(Gods 3) All the cables came out straight into the floppy/hard disk much more direct!

(Gods 4) Aurora fitted in the shorter space with 2mm spare

(Gods 4) A mounting hole on Aurora was in exactly the right place for a stud in the case

(Gods 5) The first customer asked for the ROM slot to be fitted facing outwards. This wasn't exactly planned on MinisQL - only for mini-tower cases, but it worked. He then will have a removable RomDisq.

(Gods 5) Rapid make an extended reset switch that is the right length to within 0.5mm

(Gods 6) The cooling fan is now right beside the hot SGC board.

(Gods 7) The side mounted expansion card is a few mm higher because of Goldfire (memory juts out below the board). With the enforced reversal of slots, this expansion now sits neatly OVER the QL motherboard, and the package is therefore narrower.

(Gods 8) One of the Super Gold Card heatsink screws sits exactly over a slotted hole on the QL motherboard. Removed and screwed back in through the motherboard, the SGC can be very firmly fixed to the motherboard and aligned - no more flying cards!

(Gods 8) Now that the QL motherboard is reversed it doesn't foul the hard disk mounting frame

(Gods 9) The width of the finished mplane setup is exactly the width of an AT motherboard - making design of Son of BraQuet (Mounting plate from Qbranch) a doddle.

Roy was heard to comment 'When Tony has an idea, then reverse it!'

Now I still am not sure I believe in God(s) despite singing in church choirs practically every Sunday for over 40 years, but.....

Tony Firshman also writes:
Another plus for the QL.

I have a 3GB hard disk that failed under W98 - the fat got heavily corrupted. Even FDISK refused to work on it - boot sector problem.

le it had some sort of boot sector that FDISK recognised sufficiently but then couldn't handle. It then quit to DOS without offering any rescue options.

Ron Dunnett advised I format it under the qubide driver - that formatted a 128mb partition perfectly (and blindingly fast).

I then put it back in the IBM beast, and FDISK correctly reported no boot sector and created a new one. Everything else worked and I now have a working hard disk.

Well done Phil Borman (who wrote the Qubide ROM).



JOSHEEN MARZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany
 ☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 502014)
<http://www.j-m-s.com/smsq/> email: smsq@j-m-s.com

QD

98

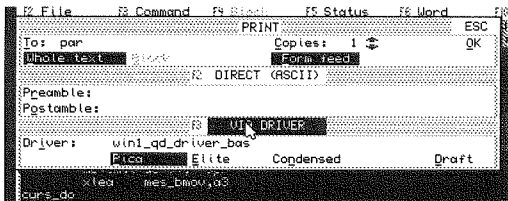
is available!

This ad does not give a list of all the features of our excellent Editor QD98, it will mainly list the new features which were introduced from the previous version QD Version 9 to the current one.

Here are its main features: "Hints" are given on menu items and the toolbar. Quite useful, because we have added many extra items and also added some additional features on "DO"ing some existing icons. Of course, this feature can be turned off.

The toolbar can be turned off and on while QD is running. A number of QL users asked me to do this because they gain two extra lines - it is done now!

Lines can be ordered, you can specify the start column in the text which is used as the sort criteria.



The GOTO label/proc/fn are now better accessible. We have also introduced a GOTO User-defined list. This means, you define a string for which QD looks and all the lines containing this text are put into the GOTO list (in fact, only the bit of the line after the string). This is very useful to create lists of all sorts, just use REMarks or other language-dependent combinations (C...)

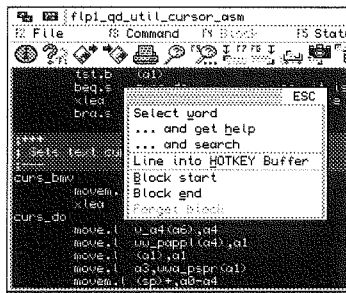
We have added scroll arrows next to every numerical value entry in all menus - you will find this very useful if you use the mouse quite a lot.

PRINTing has been completely redone. You can print as before, or via

driver. You can program the driver yourself, because it can be a BASIC filter. BASIC and Assembler examples are on the disk.

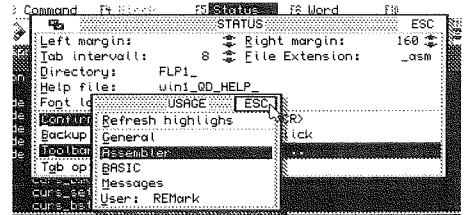
Line highlights are now possible and VERY, VERY useful. You tell QD what kind of highlights are required and it will highlight these lines in red paper. It can, for example, highlight all assembler comments (; or *), all BASIC Functions and Procedures (which

"folding"), or any first three characters editing, course, user-defined highlight all lines if you think about "Context" menu introduced. This is These are the new manual, not



is a kind of visual lines with ">" in the ters (useful for mes-PBOX etc.) and, of fined string. This will containing the string you gave - quite a neat way of doing a search, it,

which can be popped up with the right mouse button has also been much easier to handle than previous cursor-dependent functions. major changes, plus various minor ones, of course. You will get a just additional pages.



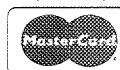
The upgrade price is DM 39,90 for owners of QD9, and DM 49,90 for owners of older versions. Please return master QD disk for upgrade. A new QD98 still costs only DM 125,-

TERMS OF PAYMENT

Postage and package [Germany] DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). [Europe] DM 14,50 (if total value of goods is up to DM 50,- then only DM 9,50). [Overseas] between DM 14,50 (1 item) and DM 35,- (maximum). All



prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.



Gee Graphics! (On the QL?) - Part 8

Building a Tetrahedron inside the QL

Herb Schaaf

Question: How do you describe a tetrahedron?

Answer: A polyhedron of four faces.

If (and only if?) you already know what a tetrahedron is, then you might be able to understand the descriptions and definitions given here. There seems to be an inverse relation between the complexity of the object and the difficulty of defining the word. The more complex polyhedron is made up of faces (which can be polygons), and those polygons in turn consist of a surface enclosed by joined segments of lines which exist between points.

What are points, lines, faces (polygons), and polyhedrons?

'Point' is one of those words that we all seem to understand, but find difficult to define. My old Webster's 5th Collegiate Dictionary gives the noun over a quarter-page of meanings, before going on to describe the verb. The word 'line' in the same dictionary uses more space, and 'face' takes even more room. 'Polygon' is a very short entry, and 'polyhedron' shorter still.

Please be patient with me. When setting up arrays in the QL to describe a tetrahedron, I'm going to use the term vertex (where 2 or more lines meet) instead of point, and use the term edge (where 2 or more faces meet) instead of line segment, and use the term face rather than polygon. This classic regular tetrahedron will have 4 vertices, 4 faces which are equilateral triangles, and 6 edges of equal length. You might say that it looks like a

pyramid with a triangular base. I want to avoid the term side since it can be ambiguous; "side" can refer to an edge of a face or to a face of a polyhedron.

Since we are visualizing our tetrahedron "inside" the QL, it helps if we think of our monitor or TV box as the place where the QL displays it for us. I usually think of the x-axis as the horizontal axis which increases in value as it goes from left to right. By the same token the y-axis is vertical and increases in value from bottom to top. I think of the z-axis as going into the monitor from near to far, or in other words from the glowing screen and increasing in value as it disappears out the back of the monitor. This arrangement is known as a left-handed system.

Left-handed or Right-handed?

To define the handedness of a system choose a hand and hold it so that: the index finger is extended parallel to your forearm, the middle finger is perpendicular to your palm, and the thumb is perpendicular to both the index and middle fin-

gers. Think of the index finger as the x-axis, the middle finger as the y-axis, and the thumb as the z-axis, with your fingernails and thumbnail pointing in the positive direction. By one convention, angles are measured around an axis of rotation by aligning the extended thumb of that hand in the positive direction of and parallel to the axis of rotation; your ring finger and little finger will naturally curl in the positive direction around that axis of rotation. There are other conventions, but I'll try to stick with this one for now.

Locating points in three dimensional (3-D) space

Imagine that x, y, and z were all equal to zero; this can be represented as a point described by the triplet (0, 0, 0). Imagine this point being located inside the middle of the monitor box so that it would be halfway from left to right, halfway from bottom to top (if you said halfway from top to bottom that would be a right-handed system), and halfway from front(near) to back(far).

HSM Coxeter's book "Regular Polytopes" has tables and values for many interesting geometric solids; we can use the following DATA set from page 52 of his book for the 4 vertices (points) of a regular tetrahedron:

<u>Vertex#</u>	<u>x</u>	<u>y</u>	<u>z</u>	<u>Word description</u>
1	1	1	1	right, top, far
2	1	-1	-1	right, bottom, near
3	-1	1	-1	left, top, near
4	-1	-1	1	left, bottom, far

The edges are between two points at a time, all combinations of 2 points from the set of 4 points:

<u>Edge#</u>	<u>Beginning Vertex</u>	<u>Ending Vertex</u>
1	1	2
2	1	3
3	1	4
4	2	3
5	2	4
6	3	4

The 4 faces contain all the combinations of 3 vertices at a time from the set of 4 vertices:

Face#	Vertex Numbers
1	1, 2, 3
2	1, 2, 4
3	1, 3, 4
4	2, 3, 4

We could also have an array that contained the faces described as a collection of enclosing edges:

Face#	Edge Numbers
1	1, 4, 2
2	1, 5, 3
3	2, 6, 3
4	4, 6, 5

Suppose we now create a series of arrays in the QL to hold this data which describes the tetrahedron in 3-D.

We could DIM tetrahedron_vertices(4,3) since the QL allows the array name to be as long as 127 characters; to keep it simple let's abbreviate and use DIM t_v(4,3) with the 4 referring to the 4 points. Since the QL's option base is zero in arrays we actually have another zeroth "phantom point" which can come in handy. The 3 then refers to elements of the array that contain the x, y, and z coordinates respectively for each point, and again we have a "spare dimension" in the zeroth place.

We create a floating point array because we will be applying mathematical manipulations to these vertex values in order to perform rotation, translation, and scaling transformations in 3-D space.

Now we can DIM t_e_v%(6,2) as an integer array for the 6 edges, defining an edge as the segment between two points (vertices) from the array t_v. This is a list of the 2 vertices that go with each of the 6 edges. We use the % to specify these as integers and thus

use less of the QL's memory. Extending this idea we can DIM t_f_v%(4,3) for the 4 faces, defining a face as the collection of 3 connected vertices. This is a list of the 3 vertices that go with each of the 4 faces.

So far, so good, and trust me, it will get more complicated.

Other possibly useful arrays will link:

the edges that go with each vertex DIM t_v_e%(4,3),

the edges that go with each face DIM t_f_e%(4,3),

the faces that go with each edge DIM t_e_f%(6,2),

the faces that go with each vertex DIM t_v_f%(4,3).

Another handy array to have is one to hold values for the "equation of the plane" and the "normal" for each of the faces. This will use floating point values, and might end up something like:

DIM t_eqn(4,10)

Translation, rotation, and scaling.

These are the three simple transformations that are often combined with another three (shearing, reflection, and perspective) by the use of matrices. I haven't gotten that far yet, but maybe before this series is finished?

Translation is similar to PAN (x-axis) and SCROLL (y-axis) and relocates the object in space; you could think of it as sliding or shifting the object parallel to one of the 3 axes. Rotation changes the orientation of the object by revolving the object around a given axis. Scaling is a change in overall size that has an effect similar to zooming.

SuperBASIC listing "Tetrahedron_bas"

Try the listing 'Tetrahedron_bas' and see the effects of the

three simple transformations. There is a menu so you can touch the key enclosed in '['] to indicate your choice. When the prompt asks you to 'ENTER' a value, put in the number and then touch the 'ENTER' key.

[R]otations are in degrees, and any value from -570000 to +570000 or so should be OK. Try values like 15, 30, 45, 90, and 180 where you know about what to expect. Confirm to your own satisfaction that the convention described earlier for positive angles in a left-handed system is being followed.

[S]cale factors larger than 1 will enlarge the object and factors smaller than 1 will reduce the object. A negative factor inverts everything; try -1 to see the effect. A factor of zero reduces everything to a single point, and that's all you will see of it after that!

The "box" we are viewing is only a 4 by 4 space, so small values from -2 to +2 for the translations will do nicely. [T]ranslations along the z axis seem to have no effect, until you follow them with a [R]otation about the x or y axis, when you will see that a change has been made.

From the [O]riginal values try the following sequence:

[T]ranslate x 2, [S]cale .5, [R]otate y 90, [R]otate x 45

After you have oriented and located the tetrahedron, [A]nimation of a crude sort will generate a sequence of images that could be stored as a series of "frames" for a movie. I hope to get to movie-making in a later article. Start with values around 12 degrees or so per frame and then experiment with larger and smaller values to see what difference it makes.

I hope you will have seen a tetrahedron in your very own QL, and that you were able to manipulate and move it in 3 dimensions.

In the next article I hope to show more general procedures that will let us build and manipulate all 5 of the Platonic Solids in the QL.

PS: Any of our UNIX or X-window folks looked at ico and/or ico2?

```
100 REMark Tetrahedron_bas
110 REMark H L Schaaf December 9, 1998
120 REMark to go with Gee Graphics #8 in QL Today
130 :
140 initialize : draw_solid t_f_v% : transforms_demo
150 :
160 DEFine PROCedure initialize
170 set_screen : set_inks
180 read_vertex_data : make_face_vertex_list
190 find_face_centers : sort_face_zs
200 END DEFine initialize
210 :
220 DEFine PROCedure transforms_demo
230 REPEAT demo_loop
240 transform_menu
250 IF ans$=="q":EXIT demo_loop
260 CLS : draw_solid t_f_v%
270 PRINT #0; "[space bar] for menu" : PAUSE
280 END REPEAT demo_loop
290 INK 7
300 END DEFine transforms_demo
310 :
320 DEFine PROCedure set_screen
330 graspix = 1/2
340 IF VER$ == "JSU" THEN graspix = 344/549
350 IF ((VER$ == "HBA") OR (VER$ == "JSL1")) THEN graspix = 476/645
360 IF graspix == 1/2 THEN PRINT "What ROM are you using?":PAUSE
370 WINDOW 512, 200, 0, 0 : PAPER 0 : INK 7 : MODE 4
380 WINDOW #0, 512, 56, 0, 200 : PAPER #0, 0 : INK #0, 4 : CLS #0
390 horz_to_vert = 512/200 : REMark the window size in pixels
400 vert_scale = 4
410 horz_width = (vert_scale / (200-1)) * (512-1) * graspix
420 SCALE vert_scale, -horz_width/2, -vert_scale/2
430 CLS
440 END DEFine set_screen
450 :
460 DEFine PROCedure read_vertex_data
470 RESTORE 560
480 DIM t_v(4,3)
490 FOR i = 1 TO DIMN(t_v)
500 FOR j = 1 TO 3
510 READ t_v(i,j)
520 END FOR j
530 END FOR i
540 END DEFine read_vertex_data
550 :
560 REMark Original DATA set for tetrahedron from Coxeter
570 DATA 1, 1, 1
580 DATA 1, -1, -1
590 DATA -1, 1, -1
600 DATA -1, -1, 1
610 :
620 DEFine PROCedure transform_menu
630 CLS #0
640 PRINT #0;"[T]ranslation, [R]otation, [S]caling, ";
650 PRINT #0;"[O]riginal data, [A]nimation or [Q]uit ?"
660 ans$=INKEY$(-1)
670 ans_num = CODE(ans$)MOD 32
680 SELEct ON ans_num
690 = 1 : motion$ = "Spin around " :spin_around
700 = 15 : initialize
710 = 17 : CLS #0 : RETurn
720 = 18 : get_variables : rotate t_v, axis_num, amount
730 = 20 : get_variables : translate t_v, axis_num, amount
740 = 19 : zoom t_v
750 = REMAINDER : GO TO 660
760 END SELEct
770 END DEFine transform_menu
```



```

780 :
790 DEFine PROCedure get_variables
800 IF ans_num = 18 : motion$="Rotation about "
810 IF ans_num = 18 : measure$=" degrees "
820 IF ans_num = 20 : motion$="Translation along "
830 IF ans_num = 20 : measure$=" amount "
840 get_axis
850 INPUT #0;"ENTER "&measure$&" of "&motion$&ans$&" axis",amount
860 END DEFine get_variables
870 :
880 DEFine PROCedure get_axis
890 PRINT #0; motion$; "which axis; [x], [y], or [z] ?"
900 ans$ = INKEY$(-1)
910 axis_num = (CODE(ans$) MOD 32 ) - 23
920 IF (( axis_num < 1 ) OR ( axis_num > 3 )) : CLS #0 : GO TO 890
930 END DEFine get_axis
940 :
950 DEFine PROCedure translate(array, axis_num, shift_amount)
960 FOR i = 1 TO DIMN(array)
970 array(i,axis_num) = array(i,axis_num) + shift_amount
980 END FOR i
990 find_face_centers : sort_face_zs
1000 END DEFine translate
1010 :
1020 DEFine PROCedure rotate(array,axis_num,turn_amount)
1030 DIM temp(3)
1040 rot_r = - RAD(turn_amount)
1050 sumin = (axis_num MOD 3) + 1
1060 difin = (sumin MOD 3) + 1
1070 cos_r = COS(rot_r) : sin_r = SIN(rot_r)
1080 FOR i = 1 TO DIMN(array)
1090 temp(axis_num) = array(i,axis_num)
1100 temp(sumin)=array(i,sumin)*cos_r + array(i,difin)*sin_r
1110 temp(difin)=array(i,difin)*cos_r - array(i,sumin)*sin_r
1120 FOR j = 1 TO 3
1130 array(i,j) = temp(j)
1140 END FOR j
1150 END FOR i
1160 find_face_centers : sort_face_zs
1170 END DEFine rotate
1180 :
1190 DEFine PROCedure zoom(array)
1200 CLS#0
1210 INPUT#0;"Enter Scaling factor",zoom_fac
1220 FOR i = 1 TO DIMN(array)
1230 FOR j = 1 TO 3
1240 array(i,j)=array(i,j) * zoom_fac
1250 END FOR j
1260 END FOR i
1270 find_face_centers : sort_face_zs
1280 END DEFine zoom
1290 :
1300 DEFine PROCedure make_face_vertex_list
1310 DIM t_f_v%(4,3)
1320 REMark 4 faces, 3 vertices each
1330 RESTORE 1350
1340 REMark vertices ordered for counter-clockwise closure
1350 DATA 3,2,1, 1,2,4, 4,3,1, 4,3,2
1360 FOR i = 1 TO DIMN(t_f_v%)
1370 FOR j = 1 TO 3
1380 READ t_f_v%(i,j)
1390 END FOR j
1400 END FOR i
1410 END DEFine make_face_vertex_list
1420 :
1430 DEFine PROCedure find_face_centers
1440 DIM t_f_c(4,3)
1450 REMark to hold the x,y,z values for face centers
1460 REMark and to determine order for drawing each face
1470 REMark for each face
1480 FOR i = 1 TO 4
1490 DIM sumxyz(3)
1500 REMark and for each vertex in that face
1510 FOR j = 1 TO 3
1520 REMark and for each of the coordinates
1530 FOR k = 1 TO 3

```

```

1540 REMark add them together
1550     sumxyz(k) = sumxyz(k) + t_v(t_f_v%(i,j),k)
1560     END FOR k
1570     END FOR j
1580 REMark then take the average of the three values
1590     FOR k = 1 TO 3
1600         t_f_c(i,k)=sumxyz(k)/3
1610     END FOR k
1620     END FOR i
1630 END DEFine find_face_centers
1640 :
1650 DEFine PROCedure set_inks
1660 DIM nks(4)
1670 REMark use 4 colors, one per face
1680 RESTORE 1690
1690 DATA 2, 4, 7, 242
1700 REMark red, green, white, gold
1710 FOR i = 1 TO 4
1720     READ nks(i)
1730 END FOR i
1740 END DEFine set_inks
1750 :
1760 DEFine PROCedure sort_face_zs
1770 REMark arrange faces by z-value of face-centers
1780 REMark put t_f_c(i,3)'s in order, showing most positive first
1790 REMark create index array of pointers for sorting z values
1800 DIM zsort(4) : FOR i = 1 TO 4 : zsort(i) = i : END FOR i
1810 REPEAT short_sort
1820     swaps = 0
1830     FOR i = 2 TO 4
1840         IF t_f_c(zsort(i-1),3) < t_f_c(zsort(i),3) THEN
1850             swaps = swaps + 1
1860             swap zsort(i-1) , zsort(i)
1870         END IF
1880     END FOR i
1890     IF NOT(swaps) : EXIT short_sort
1900 END REPEAT short_sort
1910 END DEFine sort_face_zs
1920 :
1930 DEFine PROCedure draw_solid(f_v_array)
1940 REMark at least one face (the most distant) will always be hidden
1950 FOR i = 2 TO DIMN(f_v_array)
1960     ii = zsort(i) : INK nks(ii) : FILL 1
1970     POINT t_v(f_v_array(ii,3),1) , t_v(f_v_array(ii,3),2)
1980     FOR j = 1 TO 3
1990         LINE TO t_v(f_v_array(ii,j),1) , t_v(f_v_array(ii,j),2)
2000     END FOR j
2010     FILL 0
2020 END FOR i
2030 END DEFine draw_solid
2040 :
2050 DEFine PROCedure swap (n1, n2)
2060     n1 = n1 + n2 : n2 = n1 - n2 : n1 = n1 - n2
2070 END DEFine swap
2080 :
2090 DEFine PROCedure spin_around
2100     CLS#0
2110     get_axis
2120     INPUT#0;"ENTER Degrees per frame of animation ?",frame_shift
2130     CLS #0
2140     PRINT #0; frame_shift; CHR$(186);
2150     PRINT #0;" per frame around the ";ans$;" axis",
2160     PRINT#0;"[ESC] to stop"
2170     REPEAT spin_loop
2180         rotate t_v,axis_num,frame_shift
2190         CLS
2200         draw_solid t_f_v%
2210     REMark change the INKEY$ value for display time per frame
2220     IF INKEY$(8) == CHR$(27) : EXIT spin_loop
2230     END REPEAT spin_loop
2240 END DEFine spin_around

```

Q-CELT SOFTWARE

SOFTWARE FOR THE SINCLAIR QL AND COMPATIBLES

Dilwyn Jones Bargain Bundle.....£29.95

25% Price Reduction

**NEW YEAR
BARGAIN!**

A bargain pack of his old DJC software, mostly recently updated versions of his non-pointer driven commercial programs. Includes Basic Reporter (programming aid), Vision Mixer (fancy screens display utility), Quick Posters (text poster maker), Wordscheck, Convert-PCX (PCX graphics conversion utility), Simple Mailmerge (for Archive and EasyBase), The Cat (multi column directory lister), Image Processor (graphics program for QL), Screen Snatcher (screen picture capture system), Super Disk Labeller and Super Disk Indexer.

DEV Manager.....£14.95

Utility to help to run awkward (and even not so awkward) software from sub-directories or hard disk, especially suitable for use with systems including the DEV facility and with Qubide's SUB facility. Pointer driven.

Deskjet-A5.....£11.95

Print half-size pages side by side on HP Deskjet printers, useful for making booklets etc. Pointer driven.

Mini Graphics Printer.....£14.95

Catalogue your QL screens clipart collection with this utility which prints thumbnails by the pageful! Pointer driven.

EasyBase.....£14.95 BRAND NEW!

If you have been looking for a really simple to use database program with no bells and whistles, just really easy to use for storing names and addresses, lists etc this is the one for you. Import from/export to Archive if required.

All programs available on disk only - please specify DD or HD disk required (will be supplied on HD disks unless specified).

Payment by cheque or Eurocheque (Sterling) payable to D.Branagh, or further details from:

Darren Branagh, Q-Celt Software, The Falconry, Glenmacnass,
Glendalough, Co. Wicklow, Republic of Ireland.

Tel:(+353) 404-45319 Fax: (+353) 404-45558

Email: Q_CELT@hotmail.com

OR darrenbranagh@hotmail.com

Q-Celt can supply 2nd-User QL-Hardware such as Trump Cards, Falkenberg Keyboard I/F's, unexpanded original QL's etc. Just write or call!

New ED Drives

Mark Knight

At the recent Portishead workshop I obtained a pair of ED floppy drives from W.N. Richardson with a view to reducing the number of floppy disks needed to back up my hard disks. This article is to summarise my experiences of these drives from fitting to daily use in the short time since I obtained them. I also intend to repeat some information that may be helpful to anyone thinking of buying HD or ED drives for use with a QL. First I should describe the system into which these drives were fitted: I have a PC minitower case, A Schon PC style keyboard interface and keyboard, Q-Plane, a Gold Card and QUBIDE with two Conner hard disks, one of eighty-five megabytes and the other one hundred and eighteen megabytes. The old floppy drives are twin Sony HD drives, very quiet in operation and in the eighteen months or so since I bought them completely trouble free. The Sony drives have the usual beige front panels to match the minitower, but by the time I reached Bill he only had ED drives with black front panels. Although I would have preferred the drives to match my case the colour is a minor niggle, and of course for many QL users black would be the better choice.

The new drives are from Mitsubishi with the same model number but with slightly different revision numbers and jumper settings. One has two choices of jumper settings for the drive select, DS0 and DS1, equivalent to flp1_ and flp2_ on a QL; the other has an additional setting of DS2, equal to flp3_ on a QL. The drives are slightly heavier than the Sony mechanisms and

visibly better shielded, with metal covers on areas of the drive exposed on the Sonys. As I regard the Sony drives as very good the greater weight and visibly better construction of the Mitsubishis was a surprise. Both were set to DS1 in the factory, so after changing one to DS0 I set about opening up the minitower.

Fitting was simply a matter of removing the old drives, fitting the new ones in their places and then plugging the power leads and ribbon data cables into the equivalent sockets. Switching on and attempting to boot gave no joy, the drive lights came on and stayed on and the QL booted from the hard disk even though flp1_ contained a disk with a BOOT program on it. Attempts to obtain a directory using DIR all gave "Not found" errors and both drive lights glowed continuously.

This is usual with drives if the data cable is fitted the wrong way round, so I powered down and reversed the ribbon cable; booting from the new drives then proceeded smoothly. It is typical of so-called standardisation in the computer industry that drives from two manufacturers needed the data cable fitted with opposite alignment. Once working the new drives surprised me by being even quieter than the Sony drives; with disk drives a low noise level is one sign of good quality. Some familiar with drive history on the QL may remember that Mitsubishi drives have given trouble in the past with QL disk controllers, notably the company's early DD drives and also one batch of their ED drives, which had a bug in the built-in drive firmware. To counter the bug both the Gold Card and the Super Gold Card ROMs contain special code to work

round it, activated by the quaintly named FLP_JIGGLE command. Setting:

```
FLP_JIGGLE 0
```

...switches off the special code and makes drive access quicker, while:

```
FLP_JIGGLE 1
```

...makes the errant drives work properly at a small cost in access speed. As the bug was only present in one (admittedly large) batch of drives from Mitsubishi I kept to the FLP_JIGGLE 0 setting, assuming that Mitsubishi would have fixed the bug; this proved to be the case. Read your Gold Card or Super Gold Card manual for more information about FLP_JIGGLE, though you probably won't need it. Do not use FLP_JIGGLE 1 with DD or HD drives or with ED drives not made by Mitsubishi; it will make perfectly sound disk drives unreliable.

The first day or so of testing was fairly intensive as I had a lot of disk copying to do, not specifically for this article but because I was preparing new versions of several software products to send to beta testers. I had no ED disks but used a lot of DD and HD disks. I had no trouble reading, writing or formatting DD or HD disks.

Often users of ED and HD drives have mistaken problems with the floppy disk drivers for problems with hardware. The problems usually result from either not using FLP_JIGGLE with the bugged ED drives or from trying to format and use HD disks without taking some precautions. These problems arise because of persistent bugs in the floppy drivers for the Gold Card family of products as well as in SMSQ and SMSQ/E. This should perhaps be expected; remember the old drivers only had to deal with DD disks, single sided or

double sided, and no subdirectories; the new drivers must deal with HD, ED and DD disks as well as subdirectories and automatic detection of the disk density and drive parameters, which vary much more than they used to.

The new drivers don't just detect when the disk in the drive has been changed, they must also detect the density of the disk. When you use the DIR or WDIR command, the driver will first assume the disk has not changed; if it has it will try the same density, then ED if the disk wasn't ED, if it can't read it that way it tries HD and if that fails it will finally try a DD read, first double sided then single sided: only if all these fail will you get a "not found" or "bad or changed medium" error. As well as all this there are different step rates for stepping between tracks on a disk and startup times of drives also vary from one manufacturer to another.

Given all this it is amazing that the automated process ever works, but most of the time it works fine. On some hardware setups there can be great problems formatting HD disks properly, though there is a fix for this. If you just put in a DD disk and type the standard:

```
format flp1_
```

...command, what happens? Well first the driver tries to format the first track of the disk, track zero, as an ED track, if this fails it tries that track as an HD track and only when that has also failed will it try the DD format. Once this succeeds it will assume the rest of the disk is DD as well (seems reasonable) and continue with the format on that basis.

Obviously typing the same format command with an HD disk in the drive will try an ED format first, then try HD and when

that succeeds it should format the rest of the disk as HD and give no trouble. Formatting an ED disk should actually be quicker than the others since there is no other format to attempt first.

All this is fine in theory, and no doubt it works perfectly on the hardware Tony Tebby used when he developed the disk device drivers, but on some hardware setups it doesn't. The result is that HD disks are often not formatted correctly, though nobody seems to know why. The way round it is simply to use some rarely documented extra facilities of the FORMAT command on the Gold Card and Super Gold Card (I am assured these are present in SMSQ and SMSQ/E). Note that the facilities I am about to describe may not be present on all systems, so try them and see. If you have an early Gold Card ROM you may need to upgrade (though quite how I am not sure, contact Miracle Systems for details).

The device driver will recognise extra parameters passed to it to overrule the automatic density detection, separated by the "*" (asterisk) character like this:

```
format "flp1_*D"
```

...which makes the device driver try a DD format only and give up immediately if it fails. (Note that the quotes are needed in this instance as otherwise the "*" character is interpreted as part of a multiplication expression and as a name to format the expression becomes invalid). To specify HD use:

```
format "flp1_*H"
```

...and logically enough for ED use:

```
format "flp1_*E"
```

It is very unlikely you will ever need it but the additional parameter available is the single sided double density format,

given by:

```
format "flp1_*S"
```

If you wish to give the disk a volume name, as usual up to ten characters, it is simply inserted before the asterisk, like this:

```
format "flp1_Fractals*H"
```

...and you don't need to pad out the name so that the "*" is the eleventh character of the name, as incorrectly stated in many Gold Card/Super Gold Card manuals. I am told the SMSQ and SMSQ/E format command works in exactly the same way, though only the QL version supports ED disks. If you have trouble with HD disks then always format them this way and the trouble usually vanishes instantly. Turning off the automatic density detection in this way prevents most of the trouble with HD and ED disks. It also speeds up matters if a disk is faulty and can't be formatted as it prevents the system trying alternative formats in a determined attempt to do as requested and format the disk.

Anyway, knowing this may be one of the reasons I have never had trouble with HD disks using my old HD drives, and why I had no trouble using HD disks with the new ED drives.

After a few days I obtained some second-hand ED disks and to my surprise I couldn't get any of them to format to the full 6400/6400 sectors. I tried each disk several times but no joy; on a hunch I tried flp2_. All the disks formatted perfectly first time in flp2_ to 6400/6400 sectors. flp1_ was able to read and write these same disks but reformatting them always gave less than the full 6400 sectors.

Advice from Tony Firshman was that this was due to flp1_ being a dud. As Tony has been

building and testing all the boxed and powered drives for Bill he was able to assure me that this was the only one so far to be faulty just my luck to get the dodgy one.

The next test was obvious for me, as I recently backed up my hard disk to a set of HD disks; using The Knight Safe with compression switched on this required seven HD disks, six of which finished up completely full with very little space on the last. When I used ED disks it required four, the last of which was hardly used. The whole operation proceeded very smoothly and I was impressed by the extra speed of writing to the ED disks.

A timed test was done, loading a small animation program with a large animation file from both HD and ED disks. This confirmed the theory, taking (roughly in each case) thirty-six seconds to load from DD disk,

eighteen seconds to load from HD and nine seconds from ED. Precise timings aren't important so I didn't do further timed tests, but the improved feel of loading and saving on ED disks is significant. I did find the extra speed very noticeable when loading large files from ED disk and this was easy to get used to. It seemed to take an age waiting for the same files to load from the slow DD disks afterwards.

The extra capacity of ED disks is useful and I found I was able to pack a lot of working software onto a disk when experimenting. A copy of Turbo with source code to several programs, along with compiled versions, notes and manuals in the form of text files could be packed into the subdirectory structure on one disk and still leave plenty of space for expansion. I won't be using ED

floppy disks this way myself because my twin hard disks are even bigger and very much faster, but I tried it out for the sake of readers of this article. With over 2.2 times the capacity of HD disks ED disks are handy for my hard disk backups and that is really why I was interested.

Previously the prices of ED drives put me off, but as good quality HD drives currently cost around £20 and this batch of ED drives are just £30 I think they are worthwhile. Even if the supply of ED disks dries up completely in the next few months as rumours suggest, once I have enough for my own purposes I can use the drives with HD and DD disks for normal use. If you have been thinking of obtaining ED drives for a while now is definitely the time to do so.



Assembly Language Programming - Part 3

Norman Dunbar

An Apology

I have to apologise to all of you who were (eagerly?) awaiting part 3 of the series in the last issue of QL Today. Due to circumstances beyond my control - ie my work - I was repeatedly sent away and had absolutely no chance to put the article together. To compound matters further, I was then away on two weeks holiday in Mexico. (We left to come home on Monday and Hurricane Mitch arrived on Wednesday!).

At the point where I knew I was not going to have the article ready, I sent an email to Dilwyn at his work - unfortunately he had major problems with his work's email system and never got it! Luckily, he did get a later one sent to his home email account.

So, my apologies to all concerned, let's hope it doesn't happen again. I am actually writing this part of the article while in a hotel in Cheltenham, yes, I have been sent away again so this article is being written on a laptop.

Bugs

In part 2, page 31 of QL Today there is an incorrect instruction. At the top of the page, there is an example of the CMPI instruction. The line should read:

```
CMPI.size #data,destination and not
```

```
CMP.size #data,destination
```

On page 35, I discuss the 'Decrement and branch UNLESS condition' instructions that will be appearing in the following article (this one). This should of course read 'Decrement and branch UNTIL condition'

These errors were mine and no blame should be attached to anyone else!

More Branches

At the end of part 2, I left you with a promise that the DBcc instructions would be explained in this part, but just before we do that, there is the BSR instruction. This means 'Branch to Sub-Routine' and acts very much like GOSUB in Super-Basic (an instruction I have never used in Super-Basic, but use almost in every program in assembler - strange that.)

BSR comes in 2 sizes - byte and word. The format is:

Professional & Graphical Software

ProWesS

ProWesS is a new user environment for the QL. ProWesS is short for "PROGS Window Manager", but it is much more than that. Apart from a new window manager, it contains all the system extensions from PROGS, and is essential if you want to run programs which need these extensions.

The ProWesS reader is a major part of the package. It is a hypertext document browser. This means that text files which include formatting commands (including pictures) and possibly links to other files can be displayed and read in this program. This is used in ProWesS to read (and possibly print) the manuals, and display the help files. The hypertext documents which are used by the ProWesS reader are in HTML format, the format which is popular on Internet to display World Wide Web pages.

Another important aspect of ProWesS is the possibility to allow programs to automatically install themselves on your system, and to be able to run them without resetting the system. This means that when you get a new program, all you have to do is insert the disk and indicate "start the program in flp1", a menu option in the "utilities" button. To install a program, you indicate "install software", and the software can be added to your system. This way, you don't need to know how to write a boot file to use the multi-tasking capabilities of your computer.

ProWesS includes many programming libraries. These include syslib, an interface to the operating system, PROforma, a vector graphics system, allowing rendering both on screen and on paper (via a printer driver). The DATAdesign engine is also part of ProWesS. It is a relational database system with a bonus, as you don't even need a key field. You get a powerful record at a time data manipulation extension to the language you already use. Of course it also includes ProWesS itself, the new resolution independent window manager.

PfList

Easy to use program to create listings on any printer (especially inkjet and laser). This ProWesS application allows you to indicate the files which have to be printed. Each column contains a footer which can include the filename and filedate. The listings always allow perforation. PfList can create your listings in two columns and in landscape (or both).

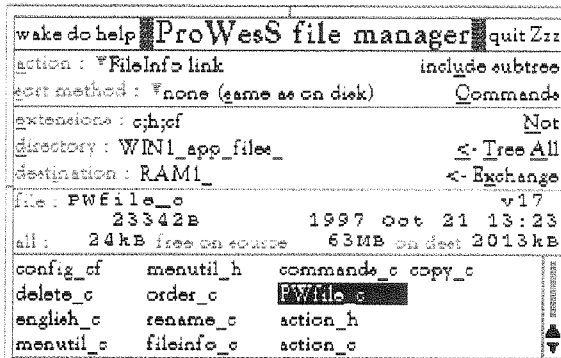
fsearch

File search utility with many useful options, like the choice to search only files with a certain extension, and whether or not the directory tree has to be scanned. All occurrences of the searchstring will be displayed with line number or offset. You can also use special matching features, like case dependent, matching a space with a stretch of whitespace, and searching for a word delimited string.

font-utills

manage your font collection. You can preview fonts on screen, see what characters exist in a font and convert Adobe Type 1 and similar fonts for use in ProWesS.

*New ProWesS application
a powerful and very user
friendly file manager*



LINEdesign

Create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can use graphics objects to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independent, meaning that the printout will be the same on any printer (e.g. same size and position).

LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. All the fonts can be displayed at any size, rotation, etc. All the fonts which are available to ProWesS can be used in LINEdesign.

LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. Lots of clipart and extra fonts are available from public domain libraries and BBS's. You can even import Adobe Illustrator files.

DATAdesign

Never before has it been so easy to create, fill in and maintain your personal databases. To start a new file, just type the names of the fields. To add or delete a field, no problem, just do it. To change the name of a field, just indicate it. You can choose which fields are displayed and also which records. You can have a hidden comment for each record, look at the file in tabulated form and transfer data to the scrap or hotkey buffer. Files can be memory based (for speed) or disk based (for safety).

new address !!

Dr. Fr. Hemerijckxlaan 13 /1
2650 Edegem
Belgium

tel : +32 (0)3/ 457 84 88
fax : +32 (0)3/ 458 62 07
email : joachim@club.innet.be
www : http://www.club.innet.be/~year2827

ProWesS - BEF 2400

DATAdesign - BEF 1200

PWfile - BEF 900

PfList - BEF 600

Payment terms :

LINEdesign - BEF 1200

fontutills - BEF 1200

fsearch - BEF 600

You have to run ProWesS to make LINEdesign, DATAdesign, fsearch, fontutills and PfList work (even though DATAdesign uses wman).

All our software is normally supplied on high density (HD) disks. However they can be obtained on double density (DD) disks at an extra costs of BEF 100. To use ProWesS and any of our other packages, you need a system with at least 2MB of memory. You should have a harddisk although a two disk system will also work. The use of SMSQ/E is strongly recommended for optimal use of ProWesS.

If you are VAT registered (specify registration number) or live outside the EEC, the amount to be paid is the total (including postage) divided by 1.21 (no need to pay too much).

Payment can be done by EuroCheque in BEF, or by VISA, EuroCard or MasterCard. Credit card orders can be handled by phone. For credit card, please specify name of card owner, card number and expiry date.

Postage : Costs of postage and packaging have to be added. You can choose the quality. Rate depends on no of programs.

copies	priority mail			ordinary mail		
	Belgium	Europe	World	Belgium	Europe	World
one	100	200	240	100	120	145
two	135	340	420	135	190	230
3 or 4	160	560	770	160	310	395
5 to 8	185	870	1250	185	550	705
more	295	1130	1610	295	800	1030

All prices are in BEF, including 21% VAT

BSR.S label Or
BSR label

Label is the destination of the subroutine to be executed. BSR is a PC relative instruction in that the destination is relative to the program counter - although it does not really look it.

The size of the instruction, byte or word, defines the size of the displacement from the PC of the following instruction to the address of label. This displacement is added to the PC and the next instruction executed is the one at that address (or PC + displacement). As the displacement is signed, the byte sized BSR can 'gosub' -128 to +127 bytes from the PC while the word sized BSR can 'gosub' -32,768 to +32,767 bytes from the PC.

At this point, a small example will maybe make things a bit clearer. Consider this chunk of (useless) code. It serves no useful purpose apart from showing the use of BSR (and a few of the other instructions we have already discussed).

Read through the following code and at the end I shall explain what it is doing. The only instruction not yet explained is RTS which for now simply means 'Return To Sender' - similar to RETURN or END DEF (sort of) in SuperBasic.

```
Start    MOVEQ    #0,D1  
Again    BSR.S    Addon  
          CMPI.L  #10,D1  
          BNE.S   Again  
          MOVEQ   #0,D0  
          RTS  
  
Addon    ADDQ.L  #1,D1  
          RTS
```

The code starts by setting D1 to zero in all 32 bits - it is a long sized move. The label 'Start' simply identifies the start of the code fragment and need not be called start - it could be called fred. It acts like a line number in SuperBasic.

The second line of code calls a sub-routine called 'addon' which lives only a few bytes further on - for this reason the byte sized variant of BSR is used and this makes the program smaller and slightly quicker - as explained later. Had the distance to the sub-routine been more than 127 bytes (or less than -128) then the assembler would have complained and the source would have had to have been amended to remove the 's' from the instruction.

The second line also has a label - 'Again'. Labels are used in assembler programs to mark significant places in the code. In SuperBasic every line must have a number - in assembler only those referenced in the code need have one, but there is no problem putting labels where it makes the code more readable.

Following on, there is a check to see if the value in D1.L is 10 (decimal) followed by a branch if not equal zero (BNE.S) to the label 'Again'. If the value in D1 is not 10 the Zero flag will not have been set and so the code will start executing from the label 'Again'. If D1.L does equal 10 then the branch to Again will be ignored.

The next line sets D0.L to zero. This is because any code that runs on a QL either as a result of a CALL address or EXECing a file returns any error codes to QDOS in D0.L and zero shows that no error has taken place. All this will be explained in a later article.

The **RTS** instruction ends a subroutine and means return to where you came from (almost). If the above code - beginning at 'Start' was called from SuperBasic, the RTS would return us to SuperBasic. If it was called from some other part of the assembler program, it would return us to the next instruction in that program.

The subroutine called from the second line begins at the label 'Addon'. It is very simply and adds 1 to the value in D1.L before the RTS returns to the place where it was called from.

Put simply, the code above loops around adding 1 to D1.L until such time as D1.L equals 10. At this point the code returns to wherever it was called from.

This is not quite true. The RTS instruction returns back to the instruction that follows the BSR one. So the above code returns to execute the CMPI.L #10,D1 instruction after running the code in the Addon subroutine.

Now that we have a few more instructions under our belts, there will be more bits of code appearing in the rest of the series. This allows the reader to alleviate the boredom of these articles and allows me to illustrate some examples of what I am trying to say!

For D0 = 10 to -1 step -1 ...

Looks a bit like SuperBasic that, but you can do the very same in assembler as well. The above code illustrating the BSR instruction can be rewritten to use the DBcc or 'Decrement and Branch' instructions. These are very similar to the Bcc instructions from part 2 of the series but they have an additional purpose. They allow a loop to be executed a set number of times and also can cause an exit from the loop if a certain condition occurs while executing the loop.

It might be better if these instructions were called DBUcc as in 'Decrement and Branch UNTIL condition' because that is actually what they do.

The full set of DBcc instructions is:

DBCC - Carry clear.
DBCS - Carry set.
DBEQ - Zero flag set.
DBF (or **DBRA**) - Branch false or always.
DBGE - Greater or equal.
DBGT - Greater than.
DBHI - Higher.
DBLE - Less or equal.
DBLS - Lower or same.
DBLT - Less than.
DBMI - Minus.
DBNE - Not equal (zero flag not set)
DBPL - Plus.
DBT - True. Very strange instruction, see below!
DBVC - Overflow clear.
DBVS - Overflow set.

The format of the instruction is :

DBcc Dn,label

The counter is always a data register, D0 to D7, and only the lowest word is affected. The label is specified as a 16 bit displacement from the PC to the next instruction to be executed. The displacement is, as usual, signed allowing branches of between -32,767 and +32,768 bytes.

This instruction does not affect the condition codes. They remain the same as they were before the instruction.

The operation of the instruction is in three parts:

First, the condition is tested to determine if the termination condition of the loop has been detected. This is the cc part. So a **DBCS** checks to see if carry is set. If the condition is detected, no branch will be performed and no decrement of the data register will be carried out either.

Second, if the condition is not detected, the lowest 16 bits of the data register is decremented by 1. If this results in a value of -1, then the loop is also terminated and no branch takes place.

Third, the branch is taken to the label specified. (PC relative).

Another example :

```
Start  MOVEQ  #1000,D1
      MOVEQ  #0,D2
Loop   ADDQ.L #1,D2
      CMPI.L #100,D2
      DBNE  D1,Loop

More   More code here ...
```

D1.L is initialised with 1,000 and D2.L is set to zero. Then the start of the loop (at label 'Loop') where 1 is added to D2.L. Following the addition, D2 is checked to see if it equals 100. The **DBNE** instruction checks the zero flag and if not set - therefore D2 is not equal 100 subtracts 1 from D1

and if this does not result in D1 becoming -1, branches to the label 'Loop' to go round again.

At the label 'More' how can you tell which of the two cases ended the loop? As you know, the loop is ended when the condition is detected or the counter reaches -1. As the **DBcc** instructions do not change the flags you can make a simple check on the Zero flag or test D1 to see if it is -1 or not. So the code that goes in at label 'More' will be this:

```
More   BNE.S  Got_100
Not_100 Process D1 = -1 here
      :
Got_100 Process D2 = 100 here
      :
```

Obviously, if we run a loop 1001 times where D1 goes from 1000 to -1, adding 1 to D2 then at some point D2 must equal 100 and that will be the only termination of the loop. D1 will never get to -1.

There are two 'interesting' **DBcc** instructions. These are 'DBF' (Decrement and Branch Until False) and 'DBT' (Decrement and Branch Until True). What is so interesting about these two?

DBF is commonly written as **DBRA** which is more meaningful as it implies that a decrement will be done followed by a branch. This is exactly what happens. The condition **FALSE** can never be created so the instruction always branches until the counter becomes -1.

DBT is the opposite. It never branches because the condition is always detected. I have never seen a **DBT** instruction used in any program I have read, written or disassembled.

Note that the loop is terminated when the counter becomes set to -1. This means that the above loop will have 1,001 iterations assuming that D2 never became 100. This can cause confusion to programmers used to processors that stop at zero. I learned on a Z80 (Sinclair ZX81) and there was a **DJNZ** instruction which subtracted 1 from the B register and branched if it was non zero.

To loop around 10 times you set B to 10 and just did it. On the 68000 series, you would set the counter to 9 not 10. Some programmers do this and others do it with the counter set to 10 but skip the first iteration. The following two examples are doing the same thing:

Example 1.

```
Start  MOVEQ  #10,D0
      BRA.S  Skip
Loop   BSR    Useful_code
Skip   DBRA  D0,Loop
```

Example 2.

```

Start  MOVEQ   #9,D0
Loop   BSR    Useful_code
      DBRA   D0,Loop

```

In example 1 the programmer sets the counter to the number of times the loop is to be executed but then skips over the loop code itself to the end of the loop. The counter is reduced to 9 and the loop is entered properly this time. The subroutine at label 'Useful_code' will be executed when the counter has values 9,8,7,6,5,4,3,2,1,0 or 10 times.

In example 2 the programmer sets the counter to 9 and then executes the code as normal. Once again the loop code at subroutine Useful_code will be executed 10 times once again, with the values 9,8,7,6,5,4,3,2,1 and 0 in the counter register D0.

Which is the best to use? It's up to you. Sometimes I use the first forms and sometimes the second. As far as reading source code is concerned, I prefer the second method because you can write something like:

```

Start  MOVEQ   #10-1,D0
      :

```

Which at least shows better that the loop will be executed 10 times. Unfortunately, when you disassemble the above instruction the assembler has calculated that 10 - 1 is 9 and it has once again become :

```

Start  MOVEQ   #9,D0
      :

```

The first method, where the loop counter is initialised with the actual iteration count, then skips the loop loses out in that there is the extra BRA.S instruction which uses up 2 bytes every time it is used, and the BRA.S has to be executed as well as the jump - all of this takes time.

Counting

In the above code fragments, I introduced the ADDQ instruction to add a value to a register. There are a few arithmetic instructions covering addition, subtraction, division and multiplication.

ADD.size source,Dn or

ADD.size Dn,destination

This adds the source to the destination. The destination is overwritten but source is not affected. The size can be byte, word or long. All the flags are affected as follows:

N is set if the result is negative, cleared if not.

Z is set if the result is zero, cleared if not.

V is set if an overflow was generated, cleared if not.

C is set if a carry was generated, cleared if not.
X is set to the same value as the C flag.

Note that byte sized ADDs cannot be done if source is An. If destination is An then ADDA should be used, however, some assemblers will convert ADD Dn,An into ADDA Dn,An for you.

ADDA.size source,An

This adds the source to the address register specified. The size can only be word or long but note that regardless of the size of source, the whole of the address register is affected. Words are sign extended to 32 bits. This instruction has no effect on the condition codes.

ADDI.size #data,destination

This instruction adds immediate data to the destination. The flags are all affected as per the ADD instruction above. The size can be byte, word or long. It is not permitted to use this to add to an address register.

ADDQ.size #data,destination

This is a very quick version of the above ADDI but it can only be used to add values between 1 and 8 to the destination. The size is byte, word or long as required. This instruction is always 2 bytes long where the ADDI can be 4 or 6 bytes. Use ADDQ wherever a value between 1 and 8 is to be added.

The flags are affected as per the ADDI instruction. The difference between this and ADDI is that you can use ADDQ to add 1,2,3,4,5,6,7 or 8 to an address register. Useful in loops.

ADDX.size Dx,Dy

ADDX.size -(Ax),-(Ay)

This one adds with the X flag added as well. It is useful when adding numbers together that are more than a register long - 32 bits. If you were to write a program that used 8 bytes in memory to store a number, then you could add two of them together using ADDX.

The destination becomes set to the value source + destination + X flag.

The flags are affected as follows :

N is set if the result is negative, cleared if not.

Z is UNCHANGED if the result is zero, cleared if not.

V is set if an overflow was generated, cleared if not.

C is set if a carry was generated, cleared if not.

X is set to the same value as the C flag.

Note the Z flag. If the result is zero it will be left as it is and not changed. If the result is non zero it is cleared. For this reason the Z flag should be set before any ADDXing takes place so that at the end, the result of zero shows up by having the Z flag still set.

This instruction and the SUBX one are mostly used in multiple precision addition and subtraction routines.

ABCD Dx,Dy ABCD -(Ax),-(Ay)

This is Add Binary Coded Decimal and is almost identical to ADDX above except that the values in the source and destination are treated as BCD instead of binary. Only 8 bits of the source and destination are affected.

A quick example :

```
Start  MOVEQ  #19,D0
        MOVEQ  #3,D1
        ABCD   D0,D1
```

Assuming that the X flag is clear, this will result in D1 being set to \$22 which is the result of adding 19 and 3 in DECIMAL. The hexadecimal numbers in the register \$19 and \$03 are interpreted as decimal digits, one digit for each 4 bits. The above example is actually adding 25 and 3 to make 34 !

The flags are affected as follows :

N is undefined.

Z is UNCHANGED if the result is zero, cleared if not.

V is UNDEFINED C is set if a DECIMAL carry was generated, cleared if not.

X is set to the same value as the C flag.

The Subtraction instructions are exactly the same as the Addition flags, but subtract instead. I have listed them below, but not explained them, read the corresponding ADD instruction for details.

SUB, SUBA, SUBI, SUBQ, SUBX and SBCD.

DIVS source,Dn

This instruction divides destination by source and puts the result into destination. Source is a word size and destination is long. The operation is carried out using signed values. The size is always word.

The destination WORD is divided by the source WORD and the result put into the destination LOW WORD. The remainder is placed in the destination HIGH WORD.

Any attempt to divide by zero will cause a divide by zero exception to occur and on a standard QL this will lock up. If overflow is detected during the operation the overflow flag is set but the operation is aborted and the source and destination are unaffected.

The flags are affected as follows :

N is set if the quotient is negative, cleared otherwise. Undefined on overflow.

Z is set if the quotient is zero, cleared if not. Undefined on overflow.

V is set if division overflow is detected. Cleared otherwise.

C is always cleared.

X is never affected. (Unchanged)

For those of us with short memories or a long period since our schooldays, the quotient is the result of the division. The remainder is what is left over.

```
Start  MOVEQ  #100,D0
        MOVEQ  #9,D1
        DIVS   D1,D0
```

Results in D0 being set to \$00010009 which is 9 remainder 1. The 9 is in the lowest word while the 1 is in the highest word.

The instruction should be read as 'divide source into destination'.

DIVU source,Dn

This is identical to the above except that both operands are treated as unsigned numbers. The flags are affected as per the DIVS instruction. Although the quotient is always positive, the N flag is set to the value in the highest bit of the lower word of destination. (ie the sign bit of a 16 bit word.)

MULS source,Dn

Multiply the destination word by the source word and place the LONG result into the destination register. Both operands are treated as signed numbers.

The flags affected are :

N - set if the result is negative, cleared otherwise.

Z - set if the result is zero, cleared otherwise.

V - Always cleared.

C - Always cleared.

X - Unchanged.

MULU source,Dn

Multiply the destination word by the source word and place the LONG result into the destination register. Both operands are treated as unsigned numbers. The flags are set or cleared as per the MULS instruction. The N flag is set to bit 31 of the result.

NEG.size destination

This instruction converts the binary value in the destination to its two's complement value. This is done by subtracting the current value from zero, putting the result back into the destination and setting the flags. All the flags are affected by this instruction. The instruction can act upon byte, word or long sized values.

The flags affected are :

N - set if the result is negative, cleared otherwise.

Z - set if the result is zero, cleared otherwise.

V - set if an overflow occurred, cleared otherwise.
C - Cleared if the result was zero, set otherwise
X - Set the same as the C flag.

NEGX.size destination

Same as NEG above except the value in the X flag is also subtracted to get the final result. The flags are not affected in the same way as NEG, but as follows :

N - set if the result is negative, cleared otherwise.
Z - set if the result is zero, UNCHANGED otherwise.
V - set if an overflow occurred, cleared otherwise.
C - Set if a 'borrow' was generated, cleared otherwise.
X - Set the same as the C flag.

NBCD destination

This instruction works on byte sized values only. It is similar to NEGX above, but the values are treated as decimal and not binary. The contents of the byte at 'destination' is subtracted from zero then the current value of the X flag is subtracted as well. The result is put back into 'destination' and the flags set as follows :

N - undefined
Z - set if the result is zero, cleared otherwise.
V - undefined
C - set if a borrow was required, cleared otherwise
X - Set the same as the C flag.

The Project

At this point, we are at the end of part 3. However now is a good time for a word or two about the project mentioned back in part 1 of this series.

Now that we are starting to understand a few more instructions, we are seeing small snippets of code. This is fine for as long as we need it, but it does not allow you to become confident in writing and understanding a fairly large program. I know, I had to learn as well even if it was a long time ago.

The project is what we shall be developing as this series progresses. We are going to write a program that is very useful for assembly language programmers - a disassembler!

The project will develop QLtdis and this will be a simple and hopefully easy to understand disassembler which will be developed of a number of issues and will be a very useful addition to your arsenal of assembler programming tools.

What tools are actually required for assembly programming? I have written, in the past, a few articles for this magazine giving a brief overview of assemblers, editors, disassemblers and monitors. These are the tools that you will need.

While it would be nice to be able to write an assembler, how would we manage to assemble it

? Hand coding in hex or binary is not easy and prone to errors. Of course, I could (?) write it and give you the hex code to feed into a hex loader, but what would you learn? How to type hex numbers into a program is all, I suspect.

Ok, a disassembler is quite a big undertaking, how do we assemble it? If you remember back to part 1, I mentioned that I had hoped to use George Gwilt's GWASS assembler for the series only to find that it will not run on a 68008. I lamented this fact and asked if anyone had a 'free' assembler that would run on the standard QL's 68008 processor.

I then received an email from George Gwilt who offered his unpublished GWASSL assembler which will run on the 68008, so thanks to George, we can all have the same assembler. QL Today will be issuing the assembler on a future cover disc. If you cannot wait to get hold of a copy, it will soon be in the Quanta library.

In this series, I have not specified which of the many addressing modes are valid for each instruction, not have I specified the 'bit pattern' of the actual instruction (its hex code) there is a reason for this. It takes a lot of typing, is very boring and prone to error. George Gwilt, once again, comes to the rescue. Part of his GWDIS (a disassembler) package includes a file holding all the instructions, their op codes, the flags affected etc etc and George has also made this available to QL Today readers. Hopefully this file will also be on the cover disc.

So now we are armed with almost all we need to get the project underway except an editor. Maybe Dilwyn will find room to put QED on the cover disc as well. (Over to you Dilwyn and company!!!!)

See you in part 4.

QDOS Bugs - Part 3

Mark Knight

Last part of our series of known QL ROM Bugs

69. If buffer full error occurs in INPUT then cursor may be left on in INPUT window and getting back to command line may be impossible in AH (SYSTEM). AH JM

Fix: Be careful on these systems not to input text longer than 128 characters using input, especially on AH.

70. CAPSLOCK affects various other characters besides the alphabetic characters though it should not do so (SYSTEM). AH JM JS MG

RWAP QL SOFTWARE

All software only available on 3.5" disk
Manuals all supplied in Quill format
All programs need 256K min. unless specified

WARGAMES

War In the East MKII v1.24 (Upgrade from original only)	£10
D-DAY MKII v3.04 (The Allies Take on the Germans)	£15
Grey Wolf v1.7 (Graphical Submarine Adventure)	£10

ADVENTURES

Return to Eden v3.08 (Graphics & Text Adventure - 3 disks)	£15
Nemesis MKII v2.01 (Text Adventure)	£10
The Prawn v2.01 (Spoof Text Adventure)	(128K) £10
Horroraday v3.1 (Humorous Text Adventure)	(128K) £10
West v2.00 (Real-time Text Adventure)	(128K) £10
The Lost Kingdom of Zkul v2.01 (Real-time Text Adventure)	(128K) £10
Adventure Package 1 (Nemesis MKII, The Prawn & Horroraday)	ONLY £25
Adventure Package 2 (Return to Eden & any other adventure)	ONLY £20
Adventure Package 3 (West & The Lost Kingdom of Zkul)	ONLY £15

GAMES

Deathstrike V1.5 (Scramble Clone)	NEW	£5
Arcade Package 2 (Hoverzone & Deathstrike)	NEW	£8
Open Golf v5.19 (Golf Program - Good Graphics)	(384K)	£10
Quizmaster II Package (Original + Question Module 2)	(128K)	£7
Stone Raider II v2.00 (Boulderdash clone)	(128K)	£5
Hoverzone v1.2 (defender clone)		£5
Arcade Package 1 (Hoverzone and Stone Raider II)	ONLY	£8

UTILITIES / GENERAL INTEREST

Sidewriter v1.08 (Print spreadsheets and text files sideways)		£10
QL Genealogist v3.21 (create a family tree)	(2MB)	£25
Genealogy for Windows on PCs - ask for details		£55
Quick Mandelbrot III v3.01 (Explore the Mandelbrot Set)		£10
3D Terrain v1.23 (Displays Abacus Files as Pictures)	(196K)	£10
SToQL v1.29 (Converts ST screens to QL format)		£10
Q-Route v1.07 (PD Demo Version Route Finder Program inc map)	(796K)	£2
Q-Route Britain Map (As at 24/8/98 - needs Q-Route!)	NEW VERSION	£2
Q-Route West Yorkshire Map (As at 24/8/98)	NEW	£1
Q-Route Catalonia Map (As at 10/10/98)	NEW	£2
Flashback SE v2.03 (Upgrade from Original Only)	(128K)	£5
(Upgrade from versions sold by RWAP Software)		£3

NOW RELEASED

The SBASIC / SuperBASIC Reference Manual (this is the book everyone is talking about - 1000+ pages of essential information for the BASIC programmer, together with electronic index, example programs and public domain toolkits). The price of this book is **£40 plus post and packing** (£7 UK, £12 Eire, £22 Europe, £32 Far East & Australasia, £25 Rest of World). Why not pre-order and collect at a show near you??

HOW TO ORDER

CONTACT: Rich Mellor, 26 Ashenhurst Road, Russells Hall, Dudley, West Midlands DY1 2HH
Tel: 01384-350043

If your system can read HD or ED disks, please specify.

(Payment must be cheque in Sterling payable to R. Mellor)
(Credit Card Users can order via QBranch)

Visit our Web Page: <http://www.qbranch.demon.co.uk/rwap.html>

Send an S.A.E. and blank disk for our current catalogue

- Fix: No easy fix but usually doesn't matter.
71. Trying to send a nul file over the network crashes the transmitting QL (SYSTEM). AH JM JS MG
Fix: if a channel is opened to the network and then it is decided that the transmission is a mistake send one byte before closing the channel.
 72. Attempts to FORMAT an mdv cartridge when either motor is still spinning (e.g. verifying the last file saved etc.) then an "in use" error occurs even if all channels to the drive are closed (SYSTEM). AH JM JS MG
Fix: Very minor bug hardly likely to bother anyone now, but if you still use micordrives only attempt to format them when no mdv drive motors are active.
 73. SCROLL accepts parameters outside the documented range and produces odd (sometimes useful) results (SYSTEM). AH JM JS MG
Fix: Same as bug 68 check your listings carefully and this bug doesn't matter. Read any documentation for the bug carefully if you intend to use the effects, but I suggest not doing so as future systems may not allow your program to run properly.
 74. Very high pitched notes produced with BEEP or equivalent system calls may make other activities of the second processor unreliable, e.g. keyboard reading and serial ports (SYSTEM). AH JM JS MG
Fix: Only good fixes are Hermes chip or Super Hermes to replace the code in the second processor which is responsible for the problems.
 75. KEYROW confuses some key combinations involving CTRL and ALT, contrary to what the QL manual says (SYSTEM). AH JM JS MG
Fix: No fix, problem is built into the design of the QL and the manual is wrong.
 76. If a multitasking job starts up while SuperBASIC executes a BEEP or KEYROW the system may crash (SYSTEM). AH JM JS MG
Fix: no known fix so be careful.
 77. Most QL systems allow priorities of multitasking jobs to be set outside the official documented range of 0 to 127, usually 0 to 255 (SYSTEM). AH JM JS MG
Fix: This bug is useful, as the range of priorities from 0 to 255 allows more differential between jobs and low priority jobs will take even less processing time from the highest priority ones. It is strongly suggested that programs for distribution never use priorities outside the official 0 - 127 range as on different systems this will produce different

effects. As an example Minerva allows signed byte values (-128 to 127) to permit background tasks. The extra speed gain from using a priority of 255 is not worth giving other users compatibility problems, it is usually around 1 - 2% though it may be more on some systems.

Minerva 1.97 problems.

1. RECOL problem (SYSTEM).
RECOL in MODE 4 doesn't recolour white pixels properly unless the seventh parameter is passed, which is the yellow parameter in MODE 8:
`RECOL 0,1,2,3,4,5,6,0`
...won't recolour white pixels to black in MODE 4, while:
`RECOL 0,1,2,3,4,5,0,0`
...will work. In MODE 8 RECOL works as it should.
2. LIGHTNING SE problem (SYSTEM).
When using Lightning SE on a Minerva system you must close windows in the opposite order to opening otherwise they will leave "ghost" windows which will be visible and cleared, redrawn etc. each time MODE is used, until a channel with the same #number is opened.
Fix: This problem is fixed by loading the Pointer Environment or simply by not loading Lightning SE. There is not much point in using Lightning SE on a Minerva 1.97 anyway as it provides little extra speed over the ROM screen driver or maths functions unless you intend to use the reduced precision options. Loading just the maths functions is fine.
3. In-Line IF problem 1 (BASIC).
There are various problems with these, the most important of which is that when such a single-line structure is used PROCedure or GO SUB calls may be called more than once when they should not be. Although most programmers would suggest that GO SUB should never be used in SuperBASIC this also applies to PROCedure calls so it is a genuine problem, though it doesn't apply to FuNction calls.
If your program contains:
`11200 IF Loaded=1 THEN PRINT FileName$:ELSE :PRINT "No file":END IF:Test_FILE`
...then the PROCedure called Test_FILE will be called twice if the variable "Loaded" does equal 1!
The easy solution is don't do it, use this instead:

```

11200 IF Loaded=1 THEN PRINT FileName$:ELSE
:PRINT "No file"
11210 Test_FILE
...or better still:
11200 IF Loaded=1 THEN
11210 PRINT FileName$
11220 ELSE
11230 PRINT "No file"
11240 END IF
11250 Test_FILE

```

...which is easier to change should the programmer wish to add to it later.

4. In-line IF problem 2 (BASIC).

If you should use the even worse structure of having an in-line IF and the ELSE clause contains another IF (I can recommend a good clinical psychologist) then Minerva can become confused about which END IF belongs to which IF (and who can blame it). So:

```

11200 IF Choice=1 THEN EXEC_W "Quill":ELSE IF
Choice=3:EXEC_W "Abacus":END IF:NEXT Loop
...may not work properly (never mind the
details!)

```

Fix: One fix could be:

```

11200 IF Choice=1 THEN EXEC_W "Quill":ELSE :IF
Choice=3:EXEC_W "Abacus":END IF:NEXT Loop
...in other words, add a colon after the ELSE!

```

A better fix is don't use these ridiculous nested single line clauses and write code like this:

```

11200 IF Choice=1 THEN
11210 EXEC_W "Quill"
11220 ELSE
11230 IF Choice=2 THEN
11240 EXEC_W "Abacus"
11250 END IF
11260 END IF
11270 NEXT Loop

```

...or better still use SElect for this kind of thing!

5. DLINE to the last line of a program could crash the computer (BASIC).

Fix: Don't do it. This can be a pain if you forget, as the only way out is the reset button.

6. ERR_xx Functions return 1 if a higher error has occurred (BASIC).

Fix: No easy fix, so don't rely on these functions, use other methods of finding errors. ERNUM followed by SElect?

7. FILL bug: Minerva 1.97 has its own version of the official QL ROMs bug number 48, with small variations just to make life more interesting (SYSTEM).

Fix: The same, draw objects starting from top or bottom rather than sides.

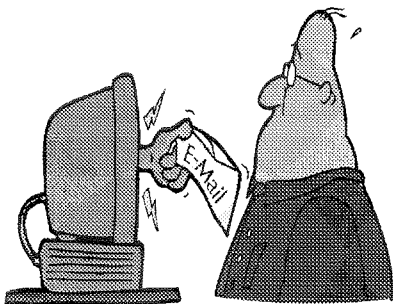
8. RENUM oddities (BASIC). Various problems are present associated with RENUM, DATA and READ, along with certain small or very large line numbers. These may result in absurd negative line numbers appearing after RENUM is used, DATA being RENUMbered and a complete system lockup, especially if you try and RUN or EDIT the program. NEW usually rescues the situation, CLEAR definitely won't.

Fix: Before reLOADing a program that exhibits this problem turn off integer tokenisation by using POKE \212,128 (and yes, do it before you LOAD). Note that QLOAD won't do in this situation, you have to LOAD a text listing type SuperBASIC file. Don't try and QSAVE a program once the problem crops up, it will probably save in such a corrupt form it will be worse than useless and will only crash your system if you QLOAD it afterwards.

9. LINE bug: Sometimes when using the SuperBASIC LINE procedure or programs that call the equivalent system routines do not join up lines that have the same floating point coordinates for their ends. This is similar to the MG ROM bug (number 58) but occurs in different circumstances (SYSTEM).

Fix: Same as the MG ROM fix, plot the end points over again using POINT or load Lightning graphics extensions.

Letter-Box



Graham Bindon writes:



This letter is on disk because I have no printer: it is about my attempt to buy

one.

My Star LC10 nine pin matrix printer has served me well for many years but it finally gave up the ghost and I thought to buy an ink jet printer. (As well as a QL I possess an ancient Acorn Archimedes which, by

courtesy of the Archimedes Fractal Group, produces the most stunning Mandelbrot images that would benefit from a high definition printer).

An Epson Stylus 440 seemed to meet my requirements of price and facility, so I bought one. You can imagine my horror when the accompanying booklet (it could not possibly be called a manual) informed me that the minimum

system requirements were a 16 megabyte PC with Windows of various kinds, a CD-ROM drive and a virtually infinite hard disk drive!

No doubt I am an innocent in such matters but one could reasonably expect that most small printers could be used with most small computers, as is the case with monitors - (at least I hope it is?).

The booklet told me how the printer was set up, how paper was changed, how to fit the ink cartridges, how to change them and what the buttons and lights were for. There was also a short bit on how to print your first document using, of course, Windows.

Assuming the booklet was a piece of commercial sheenigans aimed at selling more and bigger PCs it occurred to me that the manual containing information about printer codes and management must be missing. Then a CD unearthed itself from the packaging which I supposed contained the vital information but was inaccessible to me.

The supplier (Currys) said the printer could only run under Windows and that a 16 Mb PC was probably too small; the CD would not help me to use it otherwise.

This was confirmed by the Epson help line. Currys also said that all the makes of printer they sold had similar requirements.

All this strikes me as a shocking restraint of trade.

Can any one confirm that what I have been told is true? How do other QLers cope? I'm sure they can't all be using ancient matrix printers: in fact I know they aren't. Roy Wood's splendid print of The SBASIC/Super-BASIC Reference Manual was certainly not produced this way - nor are his colourful invoices; they are so good to look at that one quite forgets

to complain about prices. (He can't possibly be a closet PC user - can he?)

I have a vague memory of Jochen writing approvingly of an Epson printer some time ago. Perhaps they (Epson) read his article and decided that QL hoi-polois were not good for their image and made sure we couldn't use the next generation of printers.

I am in dire need of advice and instruction; can anyone out there help?

It has to be said that Currys took back the printer and refunded the price without the slightest fuss.

Although some printers describe themselves as being for Windows use only, sometimes this only extends to the instructions or the printer driver software supplied. If the printer can be used from DOS, chances are it could be used from the QL, although you may have to buy a programming manual costing almost as much as the printer itself. Other printers are truly Windows only and it can be very hard to know what to do when buying a new modern printer.

We have an article in the pipeline about learning how to program printer control codes which we hope will be of some help.

Difficulty with a printer is an all too common scenario that we hear about. Anyone out there willing to help Graham? He can be contacted at this address:

***Graham Bindon,
13 Heol-y-Felin,
Goodwick,
Pembrokeshire,
SA64 0AR, Wales.
Telephone: 01348-874258***

To repeat it once again: the cheap models are of no use for us. They have to be cheap, so the manufacturer saves on

RAM, EPROM, manual (and metal bits of the expensive models are made of plastic). The better models Stylus Color 850 and Color 740 still have the full ESC/P2 command set and can be used from non-Windows systems.

Bill McKelvey sent us an email:



You asked for feedback (about 2K), audio joke. I know we wanted new ideas for the cartoon, but this is ridiculous! - Editor!

I somewhat agree with Phil Stokes. I think we need to enhance the programs that already exist. But I do believe that we can and should change new software. The new software should have requirements listed. That way you would know what hardware it would run on. There could possibly be 2 versions, one for the plain "Black Box", another for upgraded QL (Aurora, QPC, QXL, etc.)

"AutoInstall" is also mentioned. This could be a good thing if used properly. You should have a default set-up, but you should be able to change this according to your particular set-up. Such as what drive the program is stored on, what drive the data is on, etc. This way for someone with a small amount of knowledge they could use the default, explained with doc's. An experienced user could configure it the best way for them.

Joe Hafke has the wrong idea as far as I am concerned. He is correct when he says there are inexpensive computers out there. They are called QL's! You can still purchase them at shows and from some vendors. There are still programs that are free or shareware available. (Gee, almost bundled software!) Most of the programs that are available are

easy to use and if you do have a problem a phone call, letter or even an email will result in a quick answer. You don't even have to pay extra for this service. Try that with a "BIG Blue" company, if it is free see how quick you get an answer.

Most people don't need or even use all the features that they get with their PeeCee. Most of it is just keeping up with the "Jones's". *[Shall I take that personally? - Editor]*

I use a PeeCee at my day job. It is very much a pain. It does what it wants (when it isn't crashing). The programs for ALL of the Sinclairs allow US to do what we want them to, not what they make us do.

Sorry for being long winded, but I am tired of everybody saying PeeCee this and PeeCee that. PHOOWEY



Attachments?

Ian Pizer

I will use "QL" when I mean any configuration of QL, AURORA, QXL, QPC, which is based on QDOS operating system or SMSQ. If you are reading this you must be a user of a "QL".

If you are well-informed sender or receiver of e-mail with a "QL" or PC you can probably save time by not reading what I have written below. I am writing for the amateur user of communications.

In Psychology there are 2 important words which often appear together: Attachment and Loss. Attachment being with people, animals, things, places, or situations. Loss meaning losing your teddy bear, your best friend, your sibling, your parent, your job, etc. (Reference: John Bowlby)

In "QL" and PC communications

I recently learnt about Attachments. I knew about Loss (lost messages), but what was this Attachment thing? When I use QTPI with my "QL" (I suppose my remarks also apply to other communication programs used with "QL"), I could send and receive text mail messages, thanks to my ISP (Internet Service Provider) but not Binary Files. When reading messages with QTPI the Binary files were indicated as being present but there was no feature in QTPI to transfer them to a "QL". The only way I could receive binary files was to ask the sender to encode the file with UUencode or MIME. The received encoded file contained no "Control Codes" and could be sent like normal text and decoded with the appropriate decoding program. I had blamed the deficiency on my ISP but now I realise it is a facility in QTPI which is missing.

JOCKEIN MERZ SOFTWARE 

Im stillen Winkel 12 • 47169 Duisburg • Germany

☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 14)

Ink for EPSON-Printers

Alternative supplier, tested "extremely good"!

Stylus 800	black	DM 13,90
Stylus Color	black	DM 16,90
Stylus Color	colour	DM 27,90
Stylus Color II, IIs, 820	black	DM 16,90
Stylus Color II, IIs	colour	DM 27,90
Stylus Color 500,600	black	DM 17,90
Stylus Color 500	colour	DM 27,90
Stylus Color 800, 1520	black	DM 23,90
Stylus Color 600, 800, 1520 ..	colour	DM 34,90
Stylus Color 1500	black	DM 27,90
Stylus Photo	colour	DM 36,90

Minimum order: 3 items (can be mixed)!

Ink for Canon-Printers

BJC 6...	black	DM 9,90
BJC 6...	Cyan or Yellow or Magenta ...	each DM 9,90
BJ 3...	black	DM 11,90
BJC 4...	black	DM 11,90
BJC 4...	colour	DM 17,90
BJC 8...	black	DM 14,90
BJC 8...	Cyan or Yellow or Magenta ..	each DM 14,90
BJ 30 ...	black, Pack of 3	DM 29,90
BJ 70 ...	black, Pack of 3	DM 22,90
BJ 70 ...	colour, Pack of 3	DM 27,90

Minimum order: 3 items (can be mixed)!

For postage and payment details see ad on page 13

With a PC you also need an ISP. The ISP provides the connection to the Internet or the World Wide Web and allows you to send and receive mail. When I experimented with my PC mail facility and my ISP I came across the possibility of ATTACHING a file when sending a message. The attached file being a binary file more or less without restrictions (e.g. a spreadsheet, graphics, sounds, programs etc.) or even another text file. The mail programs are so designed that you can attach any file or files you choose to a mail message and similarly you can receive attached files. Attachment adds considerably to the flexibility and power of mail. The programs encode or decode a binary file automatically using MIME (Multiple Internet Mail Extensions) or you can choose UU encoding. So the process is not dissimilar to what I described

for "QL" but it is all done without your technical intervention. Using "QL" and QTPI is so easy (and I am well familiar with it) that I will continue to use them for all but the sending or reception of binary files. The PC communication programs do a lot of the connection work for you because they include sophisticated "Script" programs designed to handle most situations. Jonathan has provided a set of Script programs so that they can be used with QTPI to simplify connections. However I have spent many hours trying to utilise his Script for my ISP (Compuserve) with notable failure. Pity! My limits were shown up there. Probably one could prepare a script program to send or receive binary files with QTPI. It would pick up your binary file, pass it through MIME then add it to your text message and Bob's your Uncle. But

unless you are a skillful programmer you may well fail too. Any programmers out there? Finally, what exactly is a binary file? A binary file consists basically of ones and zeros but can be visualised by groups of 8 binary bits (bytes) but what you find are not ASCII characters; or it can be visualised as 16 bits (hexadecimal) which also does not have a direct meaning. Some time back I had a hot discussion by e-mail with an unknown (to me) mailer called Spike H. about the difference between a binary file and a non-binary or pure text file. We ended by agreeing to disagree. Seems to be a hot subject. Another way of transferring binary files is to use a protocol called FTP (File Transfer Protocol). Presumably FTP does not need to do encoding (maybe sends the binary bits serially?).

The future-proof QL clock

Mark Knight

On reading the November/December 1998 edition of QL Today I was interested to read W. P. J. Baily's contribution about the DATE problem with QDOS and SMSQ/E. While writing my own File Manager program I ran into this problem and solved it. The problem stems from the fact that the QDOS date arithmetic routines treat the 32 bit value from the QL clock chip as an unsigned 32 bit long integer, which is fine because that is how the clock chip works. The SuperBASIC interpreter treats all 32-bit integers as signed 32-bit values. When the clock ticks up from 2,147,483,647 to 2,147,483,648 the interpreter regards it as -2,147,483,648 since the top-

most bit is treated as the sign bit and the number becomes a two's complement negative value. As far as SuperBASIC is concerned the clock now counts upwards from this value towards -1, when at 0 it will wrap around to 1960 again.

If you type:

```
print date$(2147483647)
```

...the QL will reply with:

```
2029 Jan 19 03:14:07
```

...which is correct. Typing:

```
print date$(2147483648)
```

...will produce an overflow on a JS ROM and on most other systems. This is natural as the SuperBASIC floating point structure can't store this number correctly. To obtain a correct answer type this instead:

```
print date$(-2147483648)
```

```
print date$(-2147483647)
```

...and you will obtain:

```
2029 Jan 19 03:14:08
```

```
2029 Jan 19 03:14:09
```

...which might seem a little odd but it is also correct. You are feeding the DATE\$ function signed integer values and it treats the value as an unsigned one, so the sign bit is just another bit and increases the value internally.

This proves that DATE\$ works properly, but the ridiculous thing is that DATE doesn't. It does something silly in converting the unsigned 32 bit value read from the clock chip to a floating point number, so after the clock ticks past 2029 Jan 19 03:14:07 DATE gives wrong values. Irritated, I wrote a patch called Date_FIX which replaces the DATE function with one that works; all you

have to do is load it using LRESPR or the RESPR, LBYTES and CALL sequence and your SuperBASIC or SBASIC clock routines will work properly from 1960 Jan 01 00:00:00 right up to 2097 Feb 06 06:28:15, at which point if you are still alive you are most unlikely to be bothered whether your QL clock still works.

When writing SuperBASIC or SBASIC programs you have to count from 0 up to 2,147,483,647 as the first part of the date sequence, then from -2,147,483,648 up to -1 as the second part. This also makes comparing dates awkward in SuperBASIC, but I've solved that problem too. Try this a few times:

...you won't catch it out, at least not on a JS ROM QL, and although I haven't tested it I imagine it will work in all the SuperBASIC and SBASIC environments. You are welcome to use this code to compare dates in your own programs, but be careful if typing your own version, as all the brackets and comparisons on line 150 must be exactly right for the code to work.

```

100 BORDER 1,2
110 CLS
120 FirstDate=RND * 1E9 + RND * -1E9
130 SecondDate=RND * 1E9 + RND * -1E9
140 PRINT DATE$(FirstDate),FirstDate\DATE$(SecondDate),
SecondDate
150 IF (FirstDate<0 AND SecondDate<0 AND FirstDate<SecondDate)
OR (FirstDate=0 AND (FirstDate<SecondDate OR SecondDate<0))
THEN
160 REMark FirstDate is earlier
170 PRINT "First chosen date is earlier"
180 ELSE
190 REMark SecondDate is earlier or the same
200 PRINT "Second chosen date is earlier or the same"
210 END IF

```

Now with the SuperBASIC comparison of dates working and knowing as we do that DATE\$ works, it's time to fix DATE so the assembly language is listed below for those with an assembler. Once assembled you will end up with a small file that should be loaded in your boot program using LRESPR or similar, it is 96 bytes long and once loaded DATE will work right up to 2097 Feb 06 06:28:15.

* Working version of the QL DATE function as written by Mark Knight.
 * Thanks to Simon N Goodwin in DIY Toolkit series
 * for 32-bit long integer to floating point conversion routine.

```

*
Start      MOVE.W      $110,A0      $110 = BP.INIT vector
           LEA.L      Defines,A1    Load definition address into A1
           JMP        (A0)          jump to BP.INIT it will return to caller
*
Main       MOVE.L      #$13,D0      Select MT.RCLCK routine.
           TRAP       #1            Call it to fetch 32-bit value in D1.
*
D1_Ret     MOVE.W      D1,D4         D4 will be exponent.
           MOVE.L      D1,D5         D5 will be mantissa.
           BEQ.S       Norlised      Zero is a trivial case.
           MOVE.W      #2079,D4      First guess at exponent.
           ADD.L       D1,D1         Is it normalised?
           BVS.S       Norlised      If so, return it.
           SUBQ.W      #1,D4         Not ready, halve exponent.
           MOVE.L      D1,D5         Doubled mantissa moved to D5.
           MOVEQ       #16,D0        Try 16 bit shift first.
*
Normalise  MOVE.L      D5,D1         Make copy of mantissa.
           ASL.L       D0,D1         Shift mantissa D0 places left.
           BVS.S       Too_Far       Try next shift distance.
           SUB.W       D0,D4         correct exponent for that shift.
           MOVE.L      D1,D5         New mantissa is closer.
Too_Far    ASR.W       #1,D0         Half the shift distance for next try.
           BNE.S       Normalise     Half the previous shift each time.
*
* Make sure there is enough space for the result.
*
Norlised   MOVEQ       #6,D1         Number of bytes needed is 6.
           MOVE.W      $11A,A0      Select BV.CHRIX to reserve space.
           JSR        (A0)          Call the routine

```

MOVE.L	\$58(A6),A1	Fetch safe A1 value.
SUBQ.L	#6,A1	Point to the new value.
MOVE.L	A1,\$58(A6)	update stack pointer.
MOVE.L	D5,2(A1,A6.L)	Stack the mantissa.
MOVE.W	D4,0(A1,A6.L)	Stack the exponent.
MOVEQ	#2,D4	Indicate floating point result.
MOVEQ	#0,D0	Indicate no error.
RTS		

*

Defines	DC.W	0	No PROCedures
	DC.W	0	End of PROCedure list.
	DC.W	1	One FuNction
	DC.W	Main-*	Position of code for FuNction
	DC.B	4	Length of name
	DC.B	'DATE'	Characters of name
	DC.W	0	End of FuNction list
	END		

For those without an assembler the Date_FIX file should be on the next QL Today cover disk along with the SuperBASIC listed above - oh yes, and a calendar program that works with the QL clock and runs happily from 1960 Jan 01 00:00:00 right up to (yes, you've guessed it) 2097 Feb 06 06:28:15 with Date_FIX loaded. I hope everybody's happy now - and if not contact me in 2096 if you're still able and I'll see what I can do...

UNIX-ifying QDOS

Tim Swenson

Recently a number of Unix utilities have been ported to QDOS. Given my background in Unix, I started thinking about how Unix-like one could make the QL. Using a number of utilities, could I make my QL "feel" like a Unix system?

Now before I hear cries of non-Sinclairism, let me say that Unix is my "bread and butter". I've been using and maintaining Unix systems professionally for about 10 years. Also, next to QDOS, Unix is my next favorite operating system. It's not great for desktop systems, but for servers it rocks.

The first step in my Unix-like quest is a shell. A shell, or command interpreter, is the part of an operating system that handles the interaction between the OS and the user. In MS-DOS terms, COMMAND.COM is the shell. In QDOS, the shell is really SuperBasic. In Unix there are a number of shells, Bourne, C, T, and Korn. For QDOS, Adrian Ives has refined a shell, called simply The Shell, originally started by P.J. Taylor.

The biggest advantage in using The Shell is the way that you can execute programs. QDOS does not allow command-line arguments and they have been kludged in ToolKit II. In QDOS executing a program with com-

mand-line arguments is done like this:

```
exec unzip; "flp1_file_zip"
Where as in The Shell, it is
executed like this:
```

```
unzip flp1_file_zip
```

There is no need for the exec (it is implied) and any number of arguments can come after the executable file name. This may sound trivial but it really does affect the feel of the operating system.

The Shell supports the Unix convention for creating pipes. Pipes are allowed in QDOS, but I don't think many people use them. In Unix, pipes are a vital core of the OS. An example taken from the TKII manual is:

```
EX uc, fred TO lno TO page,
ser; 'File fred at' & date$
Unix uses the vertical bar ( | ),
called the pipe sybmol, instead
of TO. An example pipe would
be:
```

```
cat file_txt | grep string |
wc -l
```

This would copy a file to a pipe, pass it to the 'grep' command and look for the word 'string' in

the file, reporting only those lines that had the string, and pass the results to 'wc' (word count) where it would count the number of lines that came from the 'grep' command. Pipes allow you to string a number of commands together, saving the need to create output files for each command. This also works kind of like programming. If need a tool to do such and such, sometimes you can string together a number of commands, via pipes, to get the same results.

The Shell also supports the background job character/command of Unix. In QDOS we have EXEC_W and EXEC, where EXEC runs the job "in the background" and returns the keyboard to SuperBasic. In Unix the difference is done with the ampersand symbol (&). If you want to run a Unix command in the background, you would end the command line with an ampersand.

The Shell uses Level-2 directories the same way the Unix does. I can 'cd bin' to get to a directory called bin. I can also 'cd /' to get the the root directory. If I am in the bin directory and I want to get into a different directory at the same level, I can do 'cd ../man' or 'cd /man'. Both commands

would take me to the same directory.

A number of Unix commands are built into The Shell. They are:

cd - change directory.

md - make a directory.

kill - Remove a job.

alias - rename or create commands.

The Shell allows for shell scripts. These are files that contain a number of Shell commands and can be used like programs and help automate some tasks.

The Shell also comes with a few external executables to work like Unix commands. 'ls' is similar to WDIR, 'lenv' is similar to the Unix 'setenv' and the QDOS ENV_LIST. 'ls' is kind of neat in that it makes QL subdirectories look like real subdirectories. If you are in a subdirectory, it displays the files in the directory, but does not append the directory name to it.

Also available from Adrian Ives's web page is RKUtils.zip, a set of utilities by Richard Kettlewell. From it comes a number of executables, with only a few having the same name as Unix commands; cat, tsort, & sort. 'more' is listed in the documentation of the RKUtils, but it is not in the distribution.

I now start trolling various free-ware distributions for any Unix-like executables. From the C68 distribution comes touch, make, cp, rm, fgrep, grep, uue, uud, tsort, sed, & diff. The primary commands here are:

touch - create a new empty file or update the date on a file.

cp - copy file(s).

rm - remove file(s).

grep - search a file(s) for a string.

From the Gnu Text Utilities comes the following commands:

wc - word count.

cut - pull characters from lines in a file.

head - display the first few lines of a file.

tail - display the last few lines of a file.

tr - translate one character for another.

uniq - output only unique lines from a file.

cmp - compare. compares two text files.

expand - convert tabs to spaces.

fold - fold long lines for finite width output device.

join - creates a union between text files.

paste - merge same lines of several files.

split - splits a file into a number of files.

unexpand - convert spaces to tabs.

sum - print checksum and block count of a file.

With these commands, I have the start of a basic Unix command set. Now I can create files (touch), delete them (rm), and copy them (cp). I just needed a way to edit text files. The standard Unix text editor is 'vi', pronounced "vee-eye" not "six" (for the Roman numerals). A clone of 'vi' elvis, has been ported to the QL. Using an 'alias' I can make elvis appear as 'vi'. The alias is:

```
^ALIAS VI=ELVIS $-
```

The \$- tells 'alias' to pass to elvis the rest of the command line. This way elvis will know which file to edit.

Another editor that can be found on Unix and QL systems is MicroEmacs. Although not specifically a Unix application, it has been ported to Unix amongst many systems.

There are also a number of utilities that are used in Unix that are in their own distribution.

Peter Tillier has ported over AWK and SED. AWK is a nice utility language for handling column data in text files. In an act of sure perversion, I did see a LISP interpreter written in AWK. This shows that the language can be fairly useful. I don't know what functional differences Peter's SED has over the one distributed with C68.

About the most important Unix utility ported to the QL so far is Perl. Perl is THE Unix utility language. It is used from writing System Administration programs, to writing CGI-bin scripts that are the data engines behind web pages. This language can practically do it all. Jonathan Hudson has answered the wish of many a QDOS/Unix user by porting Perl 4 to QDOS.

Other Unix tools that have been ported are:

PGP - Pretty Good Privacy, THE encryption program.

RCS - Revision Control System, a tool handling source code revision control.

REXX - Another utility language (don't know much about it).

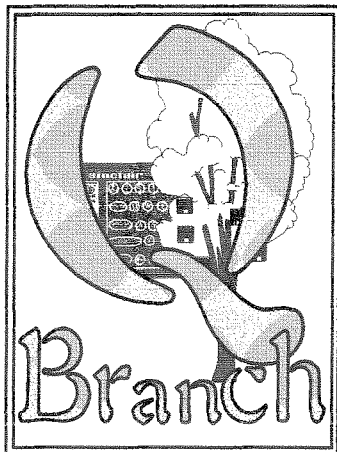
BISON & LEX - tools for creating compilers.

GZIP - Gnu ZIP, a popular compression tool.

TAR - The standard archival tools for Unix.

Curses - a C library for screen control. Curses-based programs allow portable text graphics across many terminals and systems. This library allows curses-based programs to be ported to the QL from Unix. **GhostScript** - displays Postscript & Adobe Acrobat files.

Because these programs have been compiled with C68, they have a way of knowing of they are being called by a 'shell' and then they pass back their



Shop hours :
 Mon - Fri
 10am - 6pm
 Wed / Sat
 10am - 2pm

Q Branch News

SMSQ/E
 Gold Card / Atari / QXL Version
£ 76.00
 Various Atari versions : call for details

QPC
£ 90.00 (£ 70.00 SMSQ/E Owners)
 With CUESHELL Module :
£ 110.00 (£ 90.00 SMSQ/E Owners)

We can now supply colour laptops with QPC already installed ! The systems are secondhand and vary considerably so you will need to call for details but a typical starting price for a 486 DX 70 Toshiba with 350 Meg hard disk and 12 Ram complete with QPC and Carrying case is
£ 500.00 (+postage)

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread v1.44	£ 66.00
Cueshell 2	£ 30.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00
LDUMP	£ 26.00
Q Count	£ 25.00

The Q 40 is finally here !
 By the time that you read this we should be able to take firm orders for the Q40. The package will include 16Mb Ram (upgradable to 32Mb) and an I/O card. Final pricing was yet to be decided at the time of going to press but the upper limit for the board will be £ 350.00 and it could be a lot lower if we can get the deals that we want on processors.

New Version

PROGRAMMING

QD 98	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1 & 2 (together)	£ 33.00
Easyptr pt 3 (C library)	£ 16.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

Text 87

£ 79.00

Typset 94	£ 29.00
Fountext 94	£ 39.00
2488 drivers	£ 29.00
Epson ESC/P2 drivers	£ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handling the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.
 WATCH THIS SPACE !

If you want one of these new machines then contact us. The first 25 customers to place a cheque or credit card details with us will receive free ROM upgrades and other preferential deals. No money will be cashed until the boards are ready to ship. This coming year should be a pivotal one for QDOS/SMSQ users and many new things are being planned.
 A Happy New Year to all of our customers.

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3
 £ 15.00 ea / ANY 2 PROGRAMS £ 25.00 / ALL 3 PROGRAMS £ 35.00

THESAURUS V4.01 / SOLVIT PLUS 3 v 3.00
 Pointer drivens versions of Geoff Wicks popular programs. (needs Hot_text, WMAN and PTR_GEN or SMSQ/E to run)
 Still the same price **£ 15.00**
 Upgrades from previous versions £ 2.50 + S.A.E.

Spelling Crib : PD program £ 1.50 +SAE
 or Free if you buy all three programs

You've been asking for it and here it is !

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
 compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !
£ 40.00
 + postage

Q Branch

Feeling out on a limb?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

Tel +44 (0) 1903-200212 fax +44 (0) 1903-208070
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

The Bank Volt, 6 Coronation Buildings
Ham Road, Worthing.
W. Sussex. BN11 2NN. UK.



On board Sound / Graphics / Keyboard
Interfaces

16 Mb Ram, full 68040 processor
I/O card with floppy and hard disk support

The fastest QL Hardware yet!

Special deals for the first 25 pre-orders.

Call us for details.

Hardware

QXL II	£ 180.00
SMSQ card	£ 160.00
Recycled Gold Card	£ 60.00 *
Aurora	£ 100.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
The 'Braquet'	£ 16.00

* when available.

Sold Out

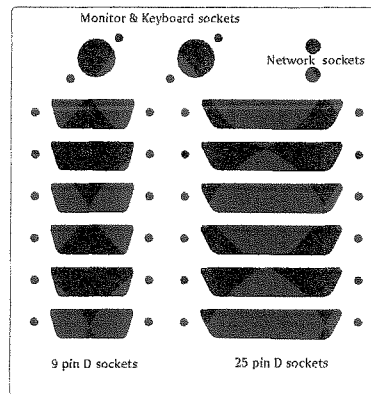
ProWesS

ProWesS	£ 48.00
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Fontpack	£ 60.00
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

Package deals

QXL II + SMSQ/E	£ 220.00
Aurora + SMSQ/E	£ 160.00
Above package + Monitor	£ 380.00

The MC - Plate



ONLY £19.95

All the socket holes you will ever need on the back of your tower case. Just rip off the silly PC thing and screw on this one. No cutting needed. What could be easier?

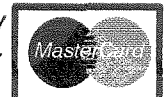
Q Branch Programs

The Knight Safe 1 - standard version	£ 30.00
The Knight Safe 2 - with compression	£ 35.00
Q - Count	£ 25.00
Pointer driven home accounting	
Q - Route v1.07	£ 25.00
Route finding programme	

2nd user 14" SVGA Colour Monitors! (suitable for Aurora)
Ready for use and giving all screen resolutions from 512 x 256 to 1024 x 576
£ 40.00 + shipping *Only a few left!*



We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.



output back to the shell to display, instead of opening up a new window. This creates a very Unix-like display.

There were some Unix commands that are not ported to the QL but I was able to find some equivalents. 'set' is equivalent to SETENV, and is called 'let' in The Shell. 'pwd' tells you what directory you are in. It can be created using the 'alias' command:

```
^ALIAS PWD=PRINT $(%DATA)
'clear' is the same as QDOS CLS and called CLS in The Shell. An 'alias' of ALIAS CLEAR=CLS will do the trick. 'mv' (move / rename) can be done by using the RKUtils command 'Ren' and give it a new name via an alias:
```

```
^ALIAS MV=REN $
```

For the Unix 'mkdir' and alias using The Shell's 'md' would be:

```
^ALIAS MKDIR=MD $
```

What I need to find now is an equivalent to the following:

ps - List jobs.

du - the equivalent of QDOS STAT.

Now, I'm not recommending that all QDOS users need to get The Shell and start using their QL's like a Unix system. What I am getting at is for those of us familiar with Unix, here is a way to work with something we are used to. Since I think in both QDOS and Unix, sometimes I find it easier to do something in Unix than I would using QDOS and non-Unix utilities. Since I bang away on a Unix system 8 hours a day, it's nice to come home and have my QL adapt itself to fit my needs. I doubt I will exclusively use The Shell to work on the QL.

Hard Disk Utilities

Dilwyn Jones

Way back in my Recursion article in Volume 1 Issue 3 I explained the subject of recursion, and one of the example routines given was an Extended DIR routine, for listing out files in directories on a hard disk.

In these example routines, I will extend this routine to provide useful little hard disk utilities. These came about because I needed utilities to provide these functions. The resultant BASIC programs are fairly

short and show just what can be achieved in a language like SuperBASIC on the QL.

1. **MAKEDIRS_bas** is a short routine which can generate a BASIC program which will recreate the directory structure of your hard disk with a series of MAKE_DIR instructions. Run the generated program before restoring a backup if your backup program does not create the subdirectories for you.

```
100 REMark extended dir of all sub-directories
110 CLS : CLS #0
120 INPUT #0,'Drive > ';dr$
130 INPUT #0,'Create program called > ';prg$
140 line_no = 100
150 OPEN_NEW #3,prg$ : REMark create the BASIC program
160 PRINT #3,line_no;' REMark reconstitute directories of ';dr$
170 PRINT line_no;' REMark reconstitute directories of ';dr$
180 line_no = line_no + 10
190 Extended_DIR dr$, ''
200 CLOSE #3 : REMark finished making the program
210 :
220 DEFine PROCedure Extended_DIR (drive$,directory$)
230 LOCAL loop,ch,d$,fp,n$
240 ch = FOP_DIR (drive$&directory$) : REMark open channel to directory
250 IF ch < 0 THEN RETURN : REMark unable to open directory
260 fp = 14 : REMark file position in directory for filename
270 REPEAT loop
280 BGET #ch\fp : IF EOF(#ch) THEN CLOSE #ch : EXIT loop
290 GET #ch,d$ : REMark get directory entry name
300 IF LEN(d$) > 0 THEN
310 REMark a directory length of 0 may be a deleted file
320 BGET #ch\fp-9 : REMark file type byte
330 IF CODE(INKEY$(#ch)) = 255 THEN
340 REMark this name is a subdirectory, so we need to DIR this
350 REMark if you want directory names printed, add this
360 REMark PRINT d$;' ->'
370 REMark write a line of BASIC
380 PRINT line_no;' MAKE_DIR ';drive$;d$
390 PRINT #3,line_no;' MAKE_DIR ';drive$;d$
400 line_no = line_no + 10
410 Extended_DIR drive$,d$
420 ELSE
430 REMark next line can be used to list all files
440 REMark PRINT d$
450 END IF
460 END IF
470 fp = fp + 64
480 END REPEAT loop
490 END DEFine Extended_DIR
```

2. **TREES_bas** prints an indented listing showing the subdirectory tree structure of your hard disk.

3. **LOCATE_bas** is a simple file finder. I placed an important QL Today file in the wrong directory on my hard disk, and thought I'd lost it. This little program quickly found it for me. As well as finding filenames for you, it can also be used to list file types - for example, to locate all Quill DOC files, just tell it to search through WIN1_, listing files ending with _DOC. Or if you know the filename, enter that as the name to search for, it'll list any directories containing files which end with the name entered.

4. **DCOPY_bas** copies an entire directory and its subdirectories, creating the directory structure elsewhere as it goes. Beware of copying long file-

```

100 REMark extended dir of all sub-directories
110 CLS : CLS #0
120 INPUT #0,'Drive > ';dr$
130 INPUT #0,'Directory > ';drc$
140 PRINT dr$
150 Extended_DIR dr$,drc$,0
160 :
170 DEFine PROCEDURE Extended_DIR (drive$,directory$,dpth)
180 LOCAL loop,ch,d$,fp,n$
190 ch = FOP_DIR (drive$&directory$) : REMark open channel to directory
200 IF ch < 0 THEN RETURN : REMark unable to open directory
210 fp = 14 : REMark file position in directory for filename
220 REPEAT loop
230 BGET #ch\fp : IF EOF(#ch) THEN CLOSE #ch : EXIT loop
240 GET #ch,d$ : REMark get directory entry name
250 IF LEN(d$) > 0 THEN
260 REMark a directory length of 0 may be a deleted file
270 BGET #ch\fp-9 : REMark file type byte
280 IF CODE(INKEY$(#ch)) = 255 THEN
290 REMark this name is a subdirectory, so we need to DIR this
300 REMark if you want directory names printed, add this
310 REMark PRINT d$; ' -,'
320 FOR a = 0 TO dpth
330 PRINT '|'; : IF a < dpth THEN PRINT ' '; : ELSE PRINT
340 END FOR a
350 FOR a = 0 TO dpth-1
360 PRINT '|'; : IF a < dpth THEN PRINT ' ';
370 END FOR a
380 PRINT '+-';d$
390 line_no = line_no + 10
400 Extended_DIR drive$,d$,dpth+1
410 ELSE
420 REMark next line can be used to list all files
430 REMark PRINT d$
440 END IF
450 END IF
460 fp = fp + 64
470 END REPEAT loop
480 END DEFine Extended_DIR

```

②

```

100 REMark extended dir of all sub-directories
110 CLS : CLS #0
120 REMark search for filename - do not enter drive name!
130 INPUT #0,'Search for > ';f$
140 INPUT #0,'On drive > ';dr$
150 INPUT #0,'From directory > ';drc$
160 Extended_DIR dr$,drc$
170 :
180 DEFine PROCEDURE Extended_DIR (drive$,directory$)
190 LOCAL loop,ch,d$,fp,n$
200 ch = FOP_DIR (drive$&directory$) : REMark open channel to directory
210 IF ch < 0 THEN RETURN : REMark unable to open directory
220 fp = 14 : REMark file position in directory for filename
230 REPEAT loop
240 BGET #ch\fp : IF EOF(#ch) THEN CLOSE #ch : EXIT loop
250 GET #ch,d$ : REMark get directory entry name
260 IF LEN(d$) > 0 THEN
270 REMark a directory length of 0 may be a deleted file
280 BGET #ch\fp-9 : REMark file type byte
290 IF CODE(INKEY$(#ch)) = 255 THEN
300 REMark this name is a subdirectory, so we need to DIR this
310 REMark if you want directory names printed, add this
320 REMark PRINT '* Searching ';d$; ' -,'
330 Extended_DIR drive$,d$
340 ELSE
350 REMark does this match f$?
360 REMark containing f$:
370 REMark IF f$ INSTR d$ : PRINT d$
380 IF LEN(d$) >= LEN(f$) THEN
390 IF d$(LEN(d$)-LEN(f$)+1 TO LEN(d$)) == f$ THEN PRINT d$
400 END IF
410 END IF
420 END IF
430 fp = fp + 64
440 END REPEAT loop
450 END DEFine Extended_DIR

```

③

names from a directory into a directory with a longer name and exceeding the QL filename length limit of 36 characters plus drive name!

These programs will compile with QLiberator. Compile with WINDS on. It might be advisable to add a line to make the program pause before it stops, so that you can view the results - adding a line such as INPUT#0,'Program Finished, press ENTER ';z\$ just before the DEFine PROCEDURE Extended_DIR line should accomplish this. I have not tested it with Turbo or Supercharge.

All programs require the use of Toolkit 2. As QDOS/SMSQ systems supporting directories tend to support the Toolkit 2 extensions, this was not thought to be a major problem.

```

100 REMark copy entire disk, subdirectories and all
110 CLS : CLS #0
120 INPUT #0, 'Copy from which drive > ';dr$
130 INPUT #0, 'Copy to which drive > ';tdr$
140 Extended_DIR dr$, ''
150 :
160 DEFine PROCedure Extended_DIR (drive$, directory$)
170 LOCAL loop, ch, d$, fp, n$
180 ch = POP_DIR (drive$&&directory$) : REMark open channel to directory
190 IF ch < 0 THEN RETURN : REMark unable to open directory
200 fp = 14 : REMark file position in directory for filename
210 REPEAT loop
220 BGET #ch\fp : IF EOF(#ch) THEN CLOSE #ch : EXIT loop
230 GET #ch, d$ : REMark get directory entry name
240 IF LEN(d$) > 0 THEN
250 REMark a directory length of 0 may be a deleted file
260 BGET #ch\fp-9 : REMark file type byte
270 IF CODE(INKEY$(#ch)) = 255 THEN
280 REMark need to create this directory in destination
290 PRINT ' Creating ';d$;'...' : MAKE_DIR tdr$&d$
300 REMark follow the directory tree
310 Extended_DIR drive$, d$
320 ELSE
330 REMark next line can be used to list all files
340 PRINT 'Copying ';d$;'...' : COPY dr$&d$ TO tdr$&d$
350 END IF
360 END IF
370 fp = fp + 64
380 END REPEAT loop
390 END DEFine Extended_DIR

```

④

The screen has a two row menu bar. The top line of the menu bar shows 'HELP' (not yet available); OPTIONS: the current file name (programme name at first) 'QUIT' and sleep buttons. The second line shows 'Files'; 'Print'; 'Edit'; 'Layout'; 'View'; each of which has sub-menus, followed by icons for selection of font; font size; colour of paper, ink and strip; underline toggle; justification; margin setting; and paragraph indenting with leading bullet ('hit' to get, 'do' to cancel) and, to the right of a page number indicator, up and down arrows for changing to next or previous page. Below this the screen shows the top part of an A4 page outlined in black with its text areas outlined in red. There is a scroll bar at the right-hand side. The pointer is in the form of a large 'I' and the cursor is a vertical line appearing initially at the start position (default left-hand margin) on the top line of TEXT #1 area.

I found to my surprise that I could not enter text directly and I have not yet found out why. However I was able to 'insert text' (Files Menu) and proceeded to do so using the 'README_first', 'changes_txt' and 'register_txt' files provided. 'README_first' was 3 pages and switching to the required page was selected by a 'hit' on the 'Page' window and again on the relevant number in a sub-menu window which appears. However the selected page appeared only on scrolling the page in the window. I was able to print the pages via the 'Print' menu (the PROforma default driver, no selection of driver yet). The other '_txt' files (one page each) also imported and printed successfully. The default font and font size seem to be 'Bitstream Charter' at 10 point.

Paragraph v1.01 - a first look

P.Bailey

This is a free initial version of a new word processor for 'QL type' computers; it is designed to operate in the 'ProWesS' environment.

I obtained a copy by downloading 'PARAG101_ZIP' from JMS BBS. The _zip file expanded into 10 files:
 smallheader._lay;
 fax._lay;
 headerfooter._lay;
 largeheader._lay;
 default._lay;
 README_first;
 register_txt;
 paragraph_exe;
 changes_txt and
 example_parg.

All but the 'paragraph_exe' file showed the warning "skipping unknown (non QDOS) field" on decompression but I pressed on. Apparently nothing was missing. The '_lay' files are page layouts and '_parg' is the (default) extension for a completed document.

After setting SYSTEM-DRAG-TEST-TIMEOUT (ProWesS configuration) to a value above 10 (I used 15 and tested 20 which appears to be too large) as instructed in 'README_first', I set up a special boot file, identical to one I use for 'LINEdesign', to give me the 'utilities' multibutton, the clock, the 'applications' multibutton and QPAC2. Programmes which could be called via the 'applications' multibutton were 'ProWesS Reader', 'PWfile', 'PARAGRAPH' and 'Editor SE'. After resetting and loading 'SMSQ/E', I ran the boot file and when I called 'PARAGRAPH' from the 'applications' multibutton the 'PARAGRAPH' screen duly appeared.

TF Services

superHermes

A major hardware upgrade for the QL

- All Hermes features (see below for list) PLUS full 19200 throughput on ser1/ser2 not affected by sound
- IBM AT keyboard interface (plus foreign drivers)
- HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps
- THREE low speed RS232 inputs (1200 to 30bps) Driver for SERIAL MOUSE supplied. Other uses include RTTY/graphics tablet etc
- THREE spare I/O lines (logic) with GND/+5V
- Capslock/scrolllock LED connector
- Turbo/keylock connectors
- 1.5k user data permanently storeable in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software) **£90** (£92/£87/£90)
 IBM AT UK layout Keyboard **£22** (£24/£23/£27)
 Serial mouse **£11** (£13/£12/£14)
 Capslock/scrolllock LED **£1** (£1.50/£1/£1.50)
 Keyboard or mouse lead **£3** (£3.50/£3/£3.50)
 High speed serial (ser3) lead **£4** (£4.50/£4/£4.50)

Hermes available for **£25** (£26/£24/£27) Working ser1/2 and independent input, debounced keyboard & keyclick.

superHermes LITE

All Hermes features (see above) + an IBM AT keyboard interface only. Entry level superHermes.

Cost (incl keyboard lead) ... **£53** (£55.50/£51/£53.50)

Minerva

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if reqd). Send disk plus SAE or two IRC's

MKI...£40 (£41/£40/£43) MKII...£65 (£66/£63/£67)

QL REPAIRS (UK only)

Fixed price for unmodified QLs. excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 including 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card. and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL. including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£41/£37/£40)
 4mbytes RomDisq..... **£65** (£66/£63/£67)
 8 mbytes RomDisq..... **£98** (£100/£95/£99)
 Aurora adaptor..... **£3** (£3.50/£3/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost..... **£34** (£36/£33/£35)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors) **£40** (£43/£38/£44)

4 amp total (for motors etc) **£45** (£48/£43/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£23/£27)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required. **£25** (£28/£23/£27)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temperature measurements, sound sampling (to 5 KHz), x/y plotting ... **£30** (£31.50/£29/£30)

Temp probe (-40°C to +125°C)..... **£10** (£10.50/£10/£11)

Connector for four temp probes **£10** (£10.50/£10/£11)

Data sheets..... **£2** (£2.50/£2/£3)

Control software & manual (for all I/F). **£2** (£2.50/£2/£3)

QL SPARES

Keyboard membrane..... **£12** (£12.50/£12/£13.50)

1377 PAL..... **£3** (£3.50/£3/£4)

Circuit diagrams..... **£3** (£3.50/£3/£4)

68008 cpu or 8049 I²C **£8** (£8.50/£7.50/£9)

8301/8302 or JM ROM or serial lead. **£10** (£11.50/£10/£11)

Power supply (sea mail overseas)..... **£12** (£17/£16/£21)

Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH! (No Eurocheques). SAE or IRC for full list and details

19 Dec 98

VISA

29 Longfield Road, TRING, Herts, HP23 4DG
 Tel: 01442-828254 Fax/BBS: 01442-828255
 tony@firshman.demon.co.uk http://www.firshman.demon.co.uk



The attributes (font etc.) can be defined before inserting the text, but the 'Total' justification attribute appears not to work at present. The original text format is maintained (line and paragraph), left-justified; the text did not flow to fill the text area as it does in 'LINEdesign'.

Using some of the available text I experimented with the various functions. I was able to use the block defining feature to change fonts and font sizes etc. within imported text. I had some difficulty in positioning the block markers correctly because (at 10 point, the default) the characters are ill-defined on the page and use of the 'View' menu which is intended to give magnification of an area always resulted in the programme quitting completely. Attempting right justification on a block of more than one line always resulted in the lowest line only being moved to the right margin.

The 'Layout' sub-menu allows definition of areas on the page for text, tables, graphics, and images (only LINEdesign pictures worked for me, _pic files appereared not to be allowed). The graphics sub_menu allows insertion of simple circles, boxes and lines (see 'example_parg'). I found that the programme became very confused if text running to more than one page was imported into a layout with more than one text area: Text Area #2 seems to take the text when Text Area #1 has been filled (probably logical but not helpful when example text areas have #2 above #1) and the second page is nonsense. This effect also seemed to lock up the programme occasionally. I managed to format my own letter head avoiding this confusion by defining a single table cell to hold my address and telephone number.

Further testing seemed to indicate that 'insert text' is sensitive to the layout defined. After loading a simple layout (one text area for the page starting 25 mm from top and ending 15.1 mm from bottom) inserting 'README_first' produced an 'insufficient memory' message in a WARNING window and stalled the programme. A machine reset was necessary to regain control.

Of course some of the problems I experienced may be system related. I use Aurora motherboard with Minerva ROM, QPLANE, Super Gold Card, Hermes, QUBIDE and hard disk, Di-Ren keyboard interface to AT keyboard, serial Mouse, and SVGA monitor.

Finally I wrote of my experiences to the author, Francois Lanciault, sending the money to register for upgrades (see registrer_.txt), by writing the letter in 'Editor' and inserting the text in the page. Yes, I am sufficiently impressed with 'PARAGRAPH' to want to have upgrades; the programme shows great promise of being developed into a very flexible word processor for our favourite computer system. I would recommend anyone who uses a word processor regularly to obtain a copy of v1.01 and examine it seriously. I am sure that most will also want to support this new venture.

[QL Today will keep you - the readers - up to date while the improvements on Paragraph carry on. The version is free, so why not give it a try - and if you like it, register just like Mr. Bailey did. If you discover problems, notify the author so that we - the QLers - can benefit from a bug-free wordprocessor as soon as possible. - Editor]

QD98

Review by Dave Westbury

We have not included any screenshots in this review because we're running out of space in this issue - please have a look at the JMS advert on page 123 for some screenshots.

Since I bought MasterSpy v1.8 back in April 1990 it has been my favourite text editor, I have always found it quick, easy to use and had just enough facilities provided for my purposes (ie, editing SuperBASIC, Assembler and binary files). Although I still find it an indispensable tool for certain things the encouraging news on the imminent (dare I use that word) arrival of new hardware and display drivers prompted me to bring myself up to date with the current QDOS/SMSQ state of the art programmers text editor QD98.

Although I had browsed the reviews of earlier QD versions in IQLR I have never used QD before so this review is very much from scratch, my only benchmark is the comparison with MasterSpy which I am very familiar with. Perhaps that may well be for the best since an article on the new features alone would only be of possible interest to existing users. Be aware that I may be a little biased to MasterSpy, no doubt my attitude may change in time as I become accustomed to QD's way of working. Also rather than detail everything provided I shall only dwell on omissions or enhancements over the basics I expect from an editor. QD98 is supplied as two DD disks and a 48 page A5 manual, the extra disk contains the HyperHelp files for SuperBASIC. The QD file is one of those files which can either be EX'ecuted directly or

LRESPR'd, the latter option just makes it a resident Thing to be used when required. It requires Menu extensions v7.50 to run which are supplied. I found the manual easy to get through.

There are three ways to tailor a working version of QD98 to the way you wish to use it (ie, Assembler, Basic etc). The QD98 file can be Configured (level 02) to set default options, you can pass parameters in a command string when you start a QD98 job and/or you can alter the options from within QD98 on-the-fly as desired. This gives the best of three worlds! Of course only meaningful options are available in each/all situations.

The command string parameter passing is probably the most useful since it allows you to pop-up customised QD's from its Thing. For example if you wish to edit a Basic program you can pass the Help directory, F10 Thing, Usage and '_bas' file extension (more on those later), similarly for Assembler. Unless I missed it, there is no way to get QD98 to automatically remove program line numbers before a passed filename is loaded and then add them on quitting, you have to invoke the menu options manually within QD98.

The display is made up of three parts; the top part contains information and loose items and a menubar, directly under that is an optional Toolbar which contains icons (picture buttons) for popular actions and below that is the main text area. The top part holds the usual move/resize/sleep items, an info item showing the current filename, a loose item for toggling insert/overwrite/protect and another info item which rather oddly is updated by a separate job. This item contains the current line and column coordi-

nates along with (when active) a flashing disk-save icon to indicate that the current file has been changed but not yet saved. My understanding of QPTR/PE is that you can only have a dynamic sprite as a pointer, not as a menu item, so this flashing must be done by the job, the manual states the line/column update is done by the separate job to avoid the overhead on the QD98 job. Similarly, a QD98 reduced to a button with unsaved text will have a flashing disk-save icon. Initially I found it a bit distracting but soon got used to it. A neat way of knowing the status of your, possibly, precious text file.

Although I said menubar it is in fact just a row of loose items which pull down menus grouping similar actions. Some of the items within these menus can be invoked directly through hitting a toolbar icon. Other icons on the toolbar perform the same action as a key (eg, F1 for help) or indeed perform actions not available elsewhere (eg, turn wordwrap on/off). Given the limited amount of space that a toolbar bar could reasonably take the icons chosen appear to give the best compromise to popular actions. Oh yes, I almost forgot, (not really Jochen!) if you pause the pointer briefly over the toolbar icon (or indeed any loose item) it pops up a little help window to tell you what the item does. As each item can invoke a different (though related) action with the left or right mouse button, the help pop-up even shows which next to a little picture of a mouse. The pause time before the help window pops up can be configured or the feature can even be removed (through a facility provided by menu_rext's system wide SET_DEFAULT command - OLE

next? just kidding!). The toolbar normally holds 19 icons but the last 5 can be replaced by an QD98 extension Thing's icons. Opting to remove the toolbar gives you two extra lines in text area.

The main text area, which is the only part which changes depth when resizing, only contains a scroll bar (no arrows), even when empty. I also expected to get a pan bar when lines became/were longer than window width, I didn't. This means to get to out-of-window text you have to use a cursor or cursor combination key. Holding down a left/right cursor key has the action of wrapping at the end of one line to the next line, not something I'm used to, nor sure if I like. The window can be split once only and although you can navigate the cursor (pointer) to the bottom half you are only allowed to write to the top half (in fact any attempt to leave the cursor in the bottom window causes it to 'bounce' back into the top half). I noticed that splitting the window had the effect of performing an undo on the current cursor line.

When loading a file into Master-Spy you have to estimate the heap size required for the file you want to load before loading it (otherwise it used the default). If you didn't give yourself enough room or you had been very productive you hit this ceiling and had to save your work, quit, and start over again with a bigger heap. QD98 doesn't impose this nonsense on you, it just gets on with it and reserves memory dynamically, a very welcome feature.

The main culture shock of QD98 for me has been getting used to the 'fluid' cursor (perhaps that isn't the right word but that's what it feels like). I am

used to the distinction (under the PE) of a pointer or a cursor. Moving the mouse usually left the cursor where it was and I got a pointer (ie, the inverse K sprite), alternately if I held down the left mouse button and moved the mouse I could forcibly drag the cursor. With QD98 all that is changed, moving the mouse now immediately moves the cursor about the window until the 'cursor' pointer goes out of the text area whereupon you get an arrow pointer. As long as the cursor is moving the line/column co-ords stay where they were, such that if you chose a menu option (by using the pointer on a pull down) which requires a cursor position (ie, to insert a character code into the file) then it will be to the cursor co-ord position you were at before you moved the cursor with the mouse. Unfortunately I couldn't find a key action which would put me back instantly at the current cursor position if didn't take a menu action which did, something I would have liked under this cursor/pointer arrangement, otherwise you need to navigate the cursor back there yourself (see below also).

Another habit I had to stop was moving the pointer into the text area and 'hitting' it with the left mouse button (ie, to 'hit' this window for keyboard input). The left mouse button now has the effect of marking the start (or end/ insert point) of a block. It would have been nice to have had block marking as left mouse button drag (as used on other systems).

One of the unfortunate side-effects of having a cursor pointer is that if you pop up another transient program the cursor takes the current pointer location on return which invariably is not where you left it (ie, navigate back again).

OK now lets try out some of QD98's features. The first test I did on QD98 was to try to use the program without actually having read the manual. Nothing is worse than a program that has actions which are not even remotely intuitive. I was happy to find that I could do quite a lot, even most of the toolbar icons looked their part (though the pop-up help soon appeared on the ones I couldn't fathom).

All of the line editing keys adopt the conventions used elsewhere and I had no trouble with them apart from the ALT+CTRL UP and DOWN combinations. I always associate any use of the CTRL key along with a cursor key as destructive, but QD98 uses ALT+CTRL+DOWN to 'insert' a line under the cursor line and ALT+CTRL+UP as sort of undo facility (MasterSpy is very flexible in letting you define the key code for each action available if you don't like the default).

I say 'sort of undo' since it only restores a line of text to what it was when it was out of the display window. Useful, but it doesn't undo my most common mistake of accidentally deleting a line. (MasterSpy was sorely missing an undo facility). An elementary word wrap option is provided, this is only effective if you are typing up to the right margin, the word (or rest of line) gets put on the next line. If you insert or delete in the middle of a line it does not reformat; acceptable for the use to which QD98 is intended (any more than that and you sometimes end up fighting the auto-reformat to maintain your layout!). Loading a file with the word wrap option on does not automatically break long lines up, they get folded at maximum line length not (word wrap) right margin.

Backspace (CTRL left) at the start of a line joins the line to the previous line but delete (CTRL right) does not do the same at the end of a line (hmm). TABs are treated as soft, spaces are inserted rather than a hard TAB code inserted (though you can insert an ASCII 9 code through a menu option). This means backspacing deletes the inserted spaces rather than the TAB in one fell swoop. SHIFT+TAB moves the cursor back by a TAB position but CTRL+SHIFT+TAB doesn't delete back to the TAB position (probably just as well, its only the ALT key away from a soft reset!). You can specify whether TABs are stored in place of white spaces for saved files.

Before moving onto the menu options I should also mention there is a useful cursor movement action which no doubt will be of use to 'C' programmers and its awful syntax. When the cursor is on a bracket character ALT+SHIFT+CTRL along with a left/right arrow key will take you to the matching closing or opening bracket. It also works on the quote marks ' and ".

Some of the following menus have options on them that can also be invoked through shortcut keys (CTRL combinations) but they only work if the pointer (cursor) is in the text window. This puzzled me at first as I thought QD98 was ignoring my CTRL X request to quit until I realised this restriction. F1 either echoes a file to screen or invokes HyperHELP (more on that later).

The F2 File pull down menu contains the usual load/save/print facilities expected plus something I have always found missing in MasterSpy, an insert file option. A nice feature is the optional maintenance of a '_bak' file on save. Also on this

SINCLAIR

Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*

MOBILE: 0850 597650

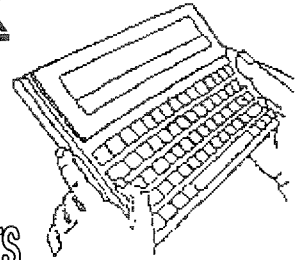
E-MAIL: WNR@COMPUSERVE.COM

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL & PC

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS (ca. 20 HOURS) *



WE DO REPAIRS, PART EXCHANGES, AND WILL BUY Z88's & PARTS

* OZ4+512k Internal Ram £150 *

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	£120	\$180
CAMBRIDGE Z88 COMPUTER	£99	\$130
☆ RECONDITIONED Z88	£60	\$90
32k RAMPACK	£18	\$25
128k RAMPACK	£38	\$50
32k EPROM PACK	£18	\$25
128k EPROM PACK	£24	\$33
256k EPROM PACK	£55	\$75
EPROM ERASER	£32	\$43
PARALLEL PRINTER LEAD	£39	\$50
SERIAL PRINTER LEAD	£12	\$16
* MAINS ADAPTOR (230vac; 6v,500ma)	£10	
TOPPER (PROTECTIVE COVER)	£12	\$16
CARRYING CASE (PLASTIC)	£10	\$12
FILE TRANSFER TO OTHER COMPUTERS		
SPECIAL Z88-QL SERIAL LEAD	£10	\$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	\$3
PCLINK KIT (For PCs) }	£25	\$35
Z88 TO MAC KIT }	£25	\$35
Z88 TO BBC KIT }	£25	\$35

**1/2 PRICE
STARTER PACK
1 - Z88 ☆ PLUS
TRANSFER KIT
+ 32K RAM
+ 32K EPROM
£72 ONLY!!**

POSTAGE UK UP TO £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

'Files' menu is an Insert Scrap option (global cut & paste buffer), I would have thought this proper to the F4 'Blocks' menu. There is a subtle feature when reading/writing DOS/TOS format disks under SMSQ/E. QD98 can auto detect the format and will add/remove CR (ASCII 13) to lines on load/save.

The F3 convention for Commands is used to pull down the menu containing some very useful features. You can not only 'Goto' to usual top/bottom/line/marker but also things like Assembler label, SuperBASIC PROC/FN definition or even a line containing a user-defined label definition. The icing on the cake is that, for example, if you have loaded a SuperBASIC program and you choose to goto a DEF PROC, you get presented with a menu containing all the DEF PROC's found in the file, picking one takes you straight to it. Although this is a great feature it unfortunately discards the list after each use, this means if there are many items you have wait for the list to be created each time.

You can set upto four markers (only one could be set in MasterSpy) to pinpoint parts of the file you may wish to quickly return to. The string Search/Replace function usefully allows to ignore case (MasterSpy was case dependent) and allows upto four strings to be stored for quick reference on future searches. (I was glad to see QD98 didn't have MasterSpy's fatal bug of locking up when asked to replace all FRED's with NEWFRED, ie, replaces on the replaced word...) There is an option to delete all control codes from a file, of limited use, but handy to extract meaningful text from a foreign format text file.

(MasterSpy allows you to strip character code ranges which is a bit more flexible).

As with MasterSpy, QD98 lets you put any character code into a file but unlike MasterSpy allows you to pick it from a character table (very handy for picking out those odd printable characters). Because QD98 appears to destroy the file integrity of binary (or executable) files it is not possible to directly alter such files as you can with MasterSpy (ie, poke in new changes). In that respect QD98 is quite different from MasterSpy since you can not index the file by byte position (Horses for courses: QD98 is good for editing source code, MasterSpy is good for object/binary code file manipulation).

Also on the Commands menu is the wonderful 'line numbers' option. Excuse me for a moment but I'd like to get up on my soap-box and shout "I HATE line numbers", thank you, I feel much better now. This, unless you didn't know, removes or adds line numbers to SuperBASIC programs. Up until I bought QD98 I used my own little program to remove them, pass the file to MasterSpy for editing and then, on quitting MasterSpy, put them back on. It would be great if a future version of QD could do this automatically on loading/saving a '.bas' file. The only argument I can possibly see for line numbers is for RESTORE. If we had some way of naming DATA blocks (eg, DATA_NAME fred, RESTORE fred) we could dispense with them altogether (GOTO & GOSUB = ZX81).

Another very useful command on the menu is 'Order lines' this simply sorts every line in your file alphabetically, ie, to get a sorted list of extensions currently available on the system: (from #0) EXTRAS\ram1_list

load it into QD98 and F3 E O. There is also a Typing check option but since I don't have QTYP I haven't tried it (I don't have much trouble with my spelling, typos - yes).

F4 invokes the Block menu, before I discuss this menu I must mention the action of the mouse with blocks. The left mouse button can either mark the start and (then) the end, or the insertion point for a block. The right mouse button (RMB) pops-up a 'context' menu (not the F4 menu). I was a little puzzled at the options on this pop-up menu, from it you can also mark the start and end of a block. What I did expect it to have it didn't - ie, copy/cut/delete/paste block options. Once you have marked your block, to do something with it you need to either select an F4 menu item, hit a toolbar item or use the direct CTRL key equivalent. I would have thought it more useful to replace the RMB pop-up menu options for block start/end with block copy/delete etc since this then doesn't require the cursor (pointer) to be moved or a CTRL key remembered. Nonetheless,

from this RMB pop-up you can cancel the block marked (which was highlighted in green ink), or, with the cursor over a word, immediately mark the word and invoke a search on it or get (Hyper)help for it. The whole line can be stuffed into the hotkey buffer from this RMB pop-up 'context' menu if you wish.

The F4 Block menu is not available until you have first marked a block, from it you can then chose to Forget/Delete/Copy/Move it, Goto Start of or Put/Add it into the 'Scrap'. The Scrap, as mentioned above, is just the global (system wide) cut & paste buffer provided by the menu_rext (QMenu) exten-

QUANTA

Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:

Monthly Newsletter - up to 40 pages

Massive Software Library - All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

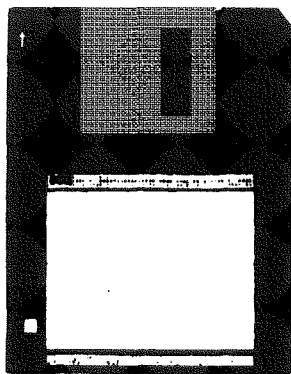
Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our FIFTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407



sions. I couldn't find a 'cut' option, so to perform one you need to put the block into the Scrap and then delete it, otherwise if you don't want to alter the scrap the block will need to be left where it is (marked) until you discard it since QD98 doesn't have a buffer local to itself. I missed the handy one-key single line cut and paste of MasterSpy, you can do the same but through two key actions (CTRL Z to put a line into the hotkey buffer then CTRL down arrow to delete the line. But when regurgitating it with ALT+SPACE you have to remember it is to/from the current cursor position unlike the CTRL Z cursor-line snapshot).

There are also a couple of other useful things you can do with blocks. You can make all the characters upper/lower case or you can shift the block left/right. This latter option is very handy because you can specify an insert string if you move the block right. For example if you make the string 'REMark' or '/*' you can comment out SuperBASIC or Assembler respectively (very useful to me).

Next on the menubar is F5 Status, from here you can customise QD98: Margins (though not line length - use Config), Tab interval, Font, Key-click, Toolbar existence, Save options (confirmation, _bak file creation) and default filing directory/file extension. Other options provided require more explanation:

'Help file' allows you to specify the current help used. If you specify a FILE then this file is echoed to the screen whenever you press F1. If you specify a DIRECTORY instead of a file then pressing F1 causes a different action. QD98 will then mark the word the cursor is

currently on and then try to open a file called 'help_index' in the specified directory. This 'help_index' file should contain a list of words it can supply help on along with the associated file that contains the help text for that word. QD98 is supplied with suitable help files for the SuperBASIC and TK2 commands such that you can get help on almost any keyword. You can add/alter these help files or even create your own help index and files. A SuperBASIC program is supplied to help create a Assembler 'help_index' file by searching for XDEF's in specified _asm file directories.

'Tab options' allows you to specify whether QD98 expands any ASCII 9 characters found in a file to equivalent white spaces (one or more spaces), also if it should try to reduce white spaces to tabs on saving the file. Upto eight different file types (as specified by file extension) can have this treatment done automatically. You have to be aware of whether your Assembler etc can digest ASCII 9, QDOS & SMSQ/E don't like them in SuperBASIC program files (many MISTakes!). Useful though for reducing file size, especially assembler.

'Usage' will, depending on the option chosen, highlight certain lines in your file (white ink on red paper). eg, for Assembler any lines marked as comment (ie, * or ;) or for SuperBASIC any DEF PROC's lines (though the manual states it does the same for DEF FN's it didn't highlight them). Alternately you can highlight lines containing a user-defined string (ie, to delimit sections).

The F6 'Word' menu offers options to interpret a marked word as a DEC/HEX/BIN number and convert it between those bases, it is slightly limited

in its use by being restricted to 16 bits. Another useful feature on this menu is being able to specify what characters can delimit a word (also definable through Config).

Finally the last item on the menubar is F10, this invokes action from a Thing that can be associated with QD98 when it is started. One of the first Things I tried was passing QD98 the SBAS/QD Thing available in SMSQ/E. This allows you to parse and execute SuperBASIC programs loaded in QD98 (including unnumbered), I had hoped it could just parse the file but it appears it always wants to run it as well. It would be nice if eventually there was also a way of parsing each line as it was entered into QD98.

I also tried the QDasm Thing by Oliver Fink supplied on the QD98 disk, although there were no instructions with it I managed to get it to invoke QMAC to assemble an '_asm' file loaded in QD98 without any problem. I imagine there is quite a potential for this F10 facility and look forward to finding more QD compatible Things, including how to write one. As mentioned above a QD extension Thing can add its own Toolbar icons if it wishes (they then replace the last five icons on toolbar which normally appear if a Thing doesn't use any).

Although QD98 comes with its own (internal) FileInfo Thing such that executing any non-executable files from QPAC Files menu will cause them to be loaded into QD98 it can be disabled through Config. This allows FileInfo 2 to still operate with the ability to pass extended command string setting-up parameters to QD98 for each file type (extension) desired. The parameters you can pass

are quite extensive - suffice to say you can set up QD98 to use template files and even pass the printer preamble/postamble strings!

On the subject of printing, in addition to the method of printing direct to a device you can opt to use a printer driver (filter program under SMSQ/E). This driver can be written in SuperBASIC or machine code etc, a simple EPSON driver example is provided in both formats. Some standard (translate) keys have already been defined and are shown in the manual, extending or changing them to cover different printers looks very easy and potentially quite powerful.

The QD98 executable file is twice the size of MasterSpy v1.8 (65K vs 34K) but packs a lot into it (both re-entrant). Whilst QD98 was not quite as fast at things like multiple search & replace it was fast enough and more than made up for it by offering case dependency. I couldn't detect any difference in speed between the two in paging up/down or navigating through a file (as long as you ignore the lazy screen option in MasterSpy - which I normally do). In the short time I have had QD98 I haven't experienced any problems with it and have happily used it to edit 120K SuperBASIC and 310K assembler files (including invoking the respective F10 Things).

So does QD98 replace MasterSpy for me? - definitely for source code files, some of the facilities available could have saved me a lot of time in the past. MasterSpy still reigns for object file manipulation but will now only be used for those occasions. As an endorsement of how easy I found QD98, I only received it two days ago and have used it to write this

review (even though it's really a programmers source editor!). Given the fact MasterSpy cost me £30 back in 1990 I think QD98 at 125DM is a bargain and was money equally well spent. (I have absolutely no connection with JMS).

[Jochen: There are some nice ideas which I may pick up in one of the future versions of QD - my reply will be slightly longer so it needs to go into the next issue. Just a quick reply to the "binary" issue: Don't try to edit

a binary file with QD - you will destroy it. QD was never intended to edit binaries, and the concept does not allow it either. QD optimises spaces at the end of a line, they are cut off. It also cuts off empty trailing lines. This means, if a byte value 32 (Space) happens to fall at the end of a line, it will be removed and the code shifts. Deadly for any binary code. Maybe a future version of QD will check the filetype...]



K-BASE - an easy to use Database

Darren D. Branagh

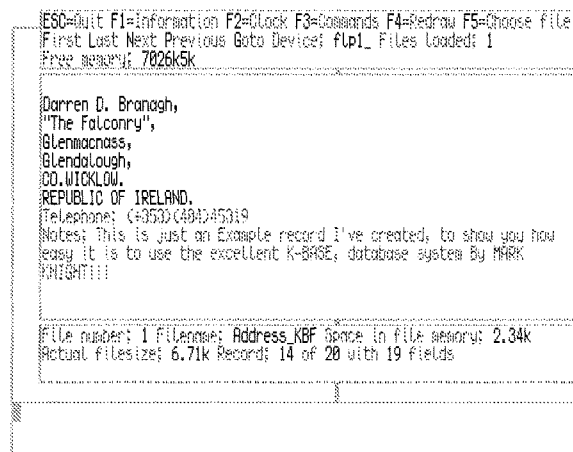
I was recently sent a bundle of Disks by our beloved QLToday editor (Cheers, Dilwyn!) with a view towards me selecting a few programs and reviewing them, for the hallowed pages of our bi-monthly magazine, as I've done so many times in the past.

What was quite different about this batch of Disks was one

table from the QUANTA library the ALARM program is particularly useful) so I was eager to Load up a few of these programs and start playing with them.

Among the disks was a program called KNIGHT SAFE (great name, Mark!) which is a Hard disk backup utility, another prog called Q-PAGE, his UTILITIES DISK,

and an unregistered version of CLIMES, his stocks and shares simulation game. However, the program I have chosen to focus this review on is K-BASE, a rather nifty little data-



thing in particular - they were all written by one very talented programmer - namely Mark Knight, from London.

I have used a couple of Marks programs before, namely via his excellent Utilities Disk (avail-

base program.

On first loading K-BASE, which is described as a "Multiple Flat File database program", you are presented with a set of neatly defined windows (it works OK in VGA mode on my QXL with

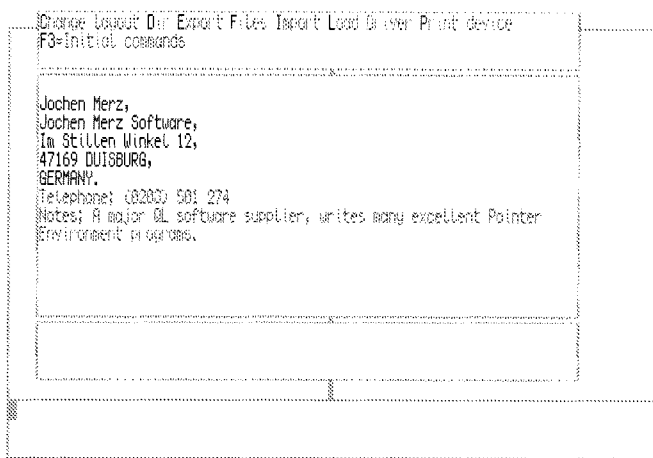
no problems, I might add) displaying the name of the Program, and the version number - 2.11 in this case. Pressing any key enters the program proper, and this is where the new defined windows come in.

The narrow window at the top of the screen is best described as a command window. It includes all the necessary key functions, i.e. ESC=Quit, F1=Info, F2=Clock, F3=Commands, F4=Redraw, etc. A similarly sized window is found at the bottom of the screen, and includes information such as the File Number, the Filename of the database that is currently in memory, the filesize in Kilobytes, and the number of fields etc. The third window, or main window as I call it, is the largest, and contains all the actual database information - actually on loading K-BASE for the first time, this contains Bill Newell, the QUANTA Membership Secretary's Address and Telephone number - as well as a Plug for QUANTA. (good Thinking, Mark!!)

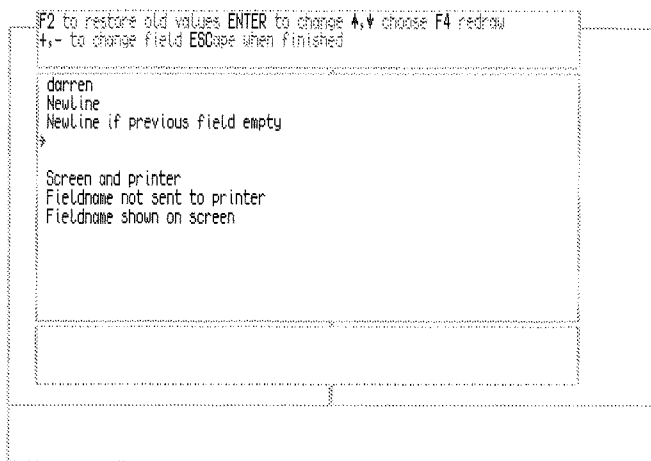
The program comes with a small sample database of some 20 records - called Address_KBF, which is useful, as it contains the addresses of many QL suppliers and traders, and you can also use this to practice on too!

Navigating through K-BASE couldn't be easier - for anyone who has ever used the QUANTA disk-based library guide program (LIBGUIDE) it will be a doddle, as its almost identical!! From the main screen, pressing F will present you with the first record in the sample database. Pressing L pulls up

the Last entry; press N for the Next one, and P for the Previous one - easy or what? To go directly to a particular entry, press G for Goto - from this point, the UP/DOWN arrow



keys with move you though K-BASE one record at a time. Using the arrow keys in combination with SHIFT will move 10 records at a time, with CTRL - 100 records, with ALT - 1,000 records, and using the UP/DOWN arrows with both SHIFT and ALT will move you through the database 10,000 (yes, 10,000!!) records at a time, though I doubt many of us with



need this feature, unless you know everyone in the Telephone directory!! You can also press F3 and then F for Find, and enter a string to search for, as a full search facility is also provided, along

with all the usual commands on F3 - Load, Save, Delete, Insert, etc..

Creating your own database from scratch is also rather easy and the program gives you an extremely wide variety of ways in which to do this. Various options are supported, such as the ability to skip any blank fields when printing, to toggle the ability to print the fieldname either on screen or to the printer or even both - stuff like this is done simply by Highlighting an option with a green arrow, and selecting it with ENTER from the F3 (Commands)

option, and by pressing B to Begin a new Database.

When you select B to Begin, you are asked for the number of Fields you require first, which is followed by the names of the fields in order which must be keyed in, along with the type of field you require - i.e. is it a word, a longword, floating point, a cash value, a date, a string - will it be in Upper Case, Lower case, or a Mixture - the choice is yours. Simply pressing the highlighted letter in each case from the choice of options will select it.

What amazes me about K-BASE is that despite its obvious complexity and flexibility, the user interface is extremely user-friendly and easy to pick up - I took to K-BASE like a duck to water, and have barely glanced at the manual (supplied in Ascii_txt format on the disk)



yet!! It's so easy to use!!

Another nice feature is the ease at which you can load several different databases and multitask them easily, without the need to load two or three programs. The number of different files loaded is displayed on the top of the screen, and they can be moved between by using the F5 (choose File) option. Other Features include the ability to check and set the clock easily by pressing F2, a rundown of all the info needed about each database you have loaded is displayed by pressing F1 such as the times and dates on which a certain database was created and last updated, and the method by which the database in question is sorted. F4 redraws (refreshes) the screen.

I've found K-BASE to be easier to use than many of the alternatives, such as ARCHIVE or FLASHBACK, (The two databases I used regularly prior to obtaining K-BASE) though these programs are probably more advanced and there is virtually nothing you can't do with them, they are difficult to use because of this - ARCHIVE in particular having an entire language of its own. K-BASE is quite flexible, yet somehow manages to retain its ease of use, and as a result, I believe it to be the best PD database program for the QL, and, unless I buy the new version of Data-Design soon to go with my copy of ProWesS, or the soon to be released EasyBase by Dilwyn Jones, or MasterBase from QBranch, I would use it. I would recommend K-BASE to anyone - it really is a great program.

Next Issue, I hope to be looking at some newly updated adventures by Rich Mellor on the QL. Stay Tuned!!

There has been a lot of talk in QL forums about trying to get other people to use the QL. We all know why we are still using after all these years and you have all, no doubt, heard me state that I still do 90% of my work on the QL in spite of the fact that I have a 200MHz Pentium laptop which many would regard as a more efficient machine. Our main problem here is one of expectation and knowledge and that is where there is a big division between us and other potential users.

I have recently been trying to guide a new user through a minefield of problems and that has highlight the problems that any new QL User would have. One of my customers runs a business importing books and his whole system is based upon several QLs all running Archive and a whole suite of complex programs written in the Archive language. I introduced him to QPC and he was very pleased to find that he could use this on a laptop to carry around his whole system. He quickly bought a few laptops for his employees and one of those has prompted this letter.

This person lives in Canada and is PC literate. At first I installed the system on his 486 laptop and this worked fine for him except for the speed. Some of the long Archive programs take up to twenty minutes to run through on this system! On his next visit to England he bought a QL/Gold Card/Disk Drives and Monitor and took it all back to Canada with him. A flurry of calls and emails between us showed

that, while he was able to handle a PC he was lost when trying to set up a QL. Why is this? Is the QL so much more complex?

Simple is Not Easy

The answer, of course, is that we have all been through the learning curve in small steps. We started off with microdrives and moved, through a variety of interfaces, to 3.5" floppies and hard disks. In that process we have all learned what we needed to know either from the magazines, user groups, books or just friendly advice. We expect a new user to get this 15 years of accumulated knowledge in a few weeks.

If you buy a new piece of PC software on either disk (rare these days since bloatware is increasingly the rule) or CD ROM you get a program called 'INSTALL' or 'SETUP' which stores all of the components of the program on the hard disk and makes changes to your system so that you can run the new program. This is needed for the PC/WINDOWS user because the whole tottering edifice of WINDOWS is not a single program but a synergy of small pieces of code all stacked together and inter-linked so that a wrong call or a piece of code in the wrong place could bring this all crashing down around you ears. In most cases this all goes fine and the system does what you expect but disaster is always lurking somewhere quite near.

The QL users simpler programs which will maybe need one or more of the standard extensions such as the Pointer

Environment, QLIB_RUN or MENU_REXT but can then be executed with a simple EXEC command if these extensions are already loaded. ProWesS has been leaning towards the PC path with regard to using '_dll' files (dynamic link libraries) and other similar concepts and Steve Hall has been probing the concepts of writing an installer program to set things up for people but this whole area is one which tends to raise the blood pressure of some people.

"You Are All Individuals", "I'm Not"

(Monty Python's Life of Brian)
We all consider ourselves to be individuals and most of us want control of the structure of our systems but, for some people, the concept of moving from a QL with a Gold Card and pair of HD or ED drives to a hard disk is terrifying and mind boggling. It is not that these people are stupid, just that they have never had to look at a boot file. Another of my customers who bought QPC (This is the pivotal point for many people - moving from a QL to QPC often means encountering a hard disk for the first time and that is where you really have to start to think about how to set the system up) has had a lot of problems. He does not want anything too fancy here. His son bought him a PC to replace his old QL and all he wanted to do was to run Quill and Archive. The way that he had done all of this before was to put the disk in the drive and reset. Faced with a system which could have all of the programs on one drive how could he select which one he wanted?

These people need a helping hand. If we are to retain a user base we have to consider those who want to do just this.

These are the people who move to a PC not because they want to perform some complex task but because the QL hardware has left them behind and the PC looks easier.

So how do we help out? I am open to suggestions here but what I would like to see is a low cost book which would lead these users from the one program per session lifestyle they have inhabited to the multitasking capability which modern systems have. Putting this into the magazine is no use at all because many of these people do not read the magazine. If anyone has any suggestions about this or would like to help writing the book contact us at Q Branch by mail, fax or email.

Sometimes I go off the rails

Let us make a comparison. The PC is like a train and the QL is a small car.

Both will go from one point to the other but the train will go only from a fixed point to another fixed point whereas the car can go from any point to any other and choose its own route. In the train you can only alter its course by buying a ticket to a different destination provided, of course, a train is going there. The car is steered by the user, many of which can drive over a bit where there is no road to get to an unusual place. In the train you get a lot of messages explaining where it is stopping but most of these are not understandable. In the car you have a set of dials which show you that you have stopped because you have run out of petrol or overheated and you can fix that.

Sometimes the railway people provide the train with a new engine to make it go faster but then couple another dozen

carriages onto it so it still goes at the same speed. It could carry more passengers but what usually happens is most of the carriages are nearly empty with each person trying to occupy two or more seats.

In the car you can decide who your passengers are, OK you may have to travel somewhere with a cranky old relative or an obstreperous teenager once in a while but in the train you may settle down with a good book in a half empty carriage only to be invaded by a bunch of football hooligans one of who is bound to pull the communication cord and stop everything for a while. Of course you may never find out which one pulled the cord and each one blames the other. If the same thing happens again on the same journey it is always someone else who does it.

The railway people also like to paint their carriages with many different colours and logos but most of us are quite happy to have our cars painted in just a few.

A car will use a small amount of fuel but that is used to travel to the places it has to go to. A train will need a large amount of fuel which will all be used up if it is full of passengers or empty because its system makes sure that it will always go from the starting point to the end point and stops at all the points in between regardless of whether anyone wants to get on or off it. This, for the car, is like driving to the supermarket and not getting out because the larder is still full of food from last week.

Trains can go much faster than cars but often stop with a lame excuse which is usually not something you can easily understand. ("Leaves on the line have performed an illegal operation and will be closed

down"). Cars do get caught in traffic jams but you can often find a way around them. Trains go on strike, cars don't.

For most people, using a train is just spending a lot of money on a ticket and being taken for a ride in uncomfortable circumstances with long waits at each end and at some points in between. To use a car you have to learn something, practise and then take a test.

could be no better indication that the job would be finished quickly. There were quite a few people at this meeting and Per Witte joined the illustrious ranks of those who bought all the components and built his Aurora system on the spot. I would like to see more of this at workshops because it demonstrates the spirit which has kept the QL going over all of these years.

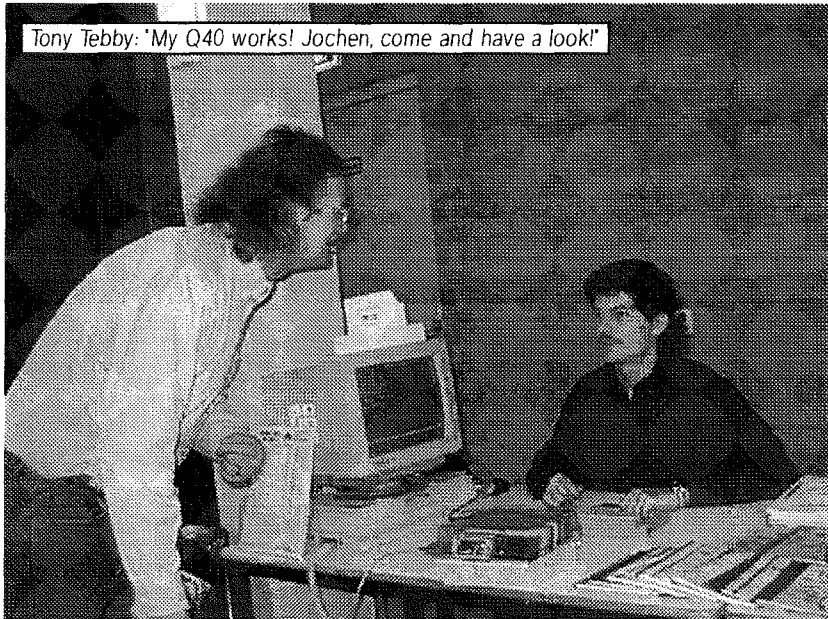
WesS. After the show we all went to the usual Chinese Restaurant and thence to a local bar to talk over all things QL.

There's Snow Business

It was never going to be an easy journey but it **could have been worse**. I set off to meet Tony Firshman at Dover and got brought to a complete standstill on the M25 as the result of an accident. Time was tight for the ferry but we made it - just. Our normal ferry journeys to foreign QL shows are taken up with a leisurely meal in the ships restaurant but Sea France does not provide a vegetarian option in their restaurant so we were forced into the self service buffet in which Tony got into an argument with the 'chef' (well he had a tall hat - and probably a pointed head) about not having Ratatouille on his steak (I am the veggie not Tony).

Ah well it could have been worse.

Off the ferry and I took over the driving duties to head off to pick up Jochen. Barely 1 hour into the drive I was overtaking a truck when the car took on a life and direction of its own. Black Ice. We performed a neat imitation of the Waltzer for a



A Dutch Treat

The last Eindhoven meeting, in November, was one of the most active I have been to. Tony Tebby was there demonstrating the latest developments in the SMSQ/E system for the Q40. Although this version of the operating system had no floppy or hard disk support at the time it was still a very rapid piece of code and the best part of this was Tony's evident enthusiasm for the project. In many ways it was this obvious keenness that led T.F. Services and I to make a joint agreement to build and sell the finished item. TT said he would give this priority because he wanted to be able to use a Q 40 for his own development work and there

Joachim van der Auwera was also there talking about his plans for the future of the Pro-WesS system. He too was very interested in buying a Q40 and had some very good concepts for the next generation of Pro-

The view from the hotel in Croatia



minute or two and wound up going backwards into the crash barrier at around 40 mph. Having ground to a halt Tony looked up with a sigh of relief only to see the truck baring down on us. Crash! The truck hit the passenger door and spun us around - well we were facing the right way now at least. We got out and begun to inspect the damage but soon became aware that vehicles were spinning to oblivion all around us. When a large truck hit the crash barrier just a few yards from us we decided that this was the time to go and park somewhere safe.

Well it could have been worse. The damage was bad but not so bad that we could not continue. Tony rewired the rear lights with a superHermes LED cable and we carried on to Jochen. At this point, we were using Tony's map and travelling up through Maastricht. He kept saying "This road is not marked on the map". We finally picked up Jochen and headed off to get Marcel Kilgus from Stuttgart. One useful item came from Jochen's house - we took his shaving mirror and taped it to what was left of the nearside wing mirror so we could see behind.

The journey down to Marcel was OK and from there we set off towards Munich and into the snow.

Well it could have been worse. The snow started just outside Munich and gradually got worse as we headed towards Austria (why was I not surprised?) and it was then that we discovered why some roads were not marked on Tony's map. Jochen noticed that the map was older than he was! This is when we had our next brush with automotive disaster. Just as we were approaching the Austrian border

the car ahead stopped dead. I applied the brakes but, with four people and a carload of QL stuff were a bit heavy and we carried on going until we were around six inches from the stationary vehicle - when we stopped.

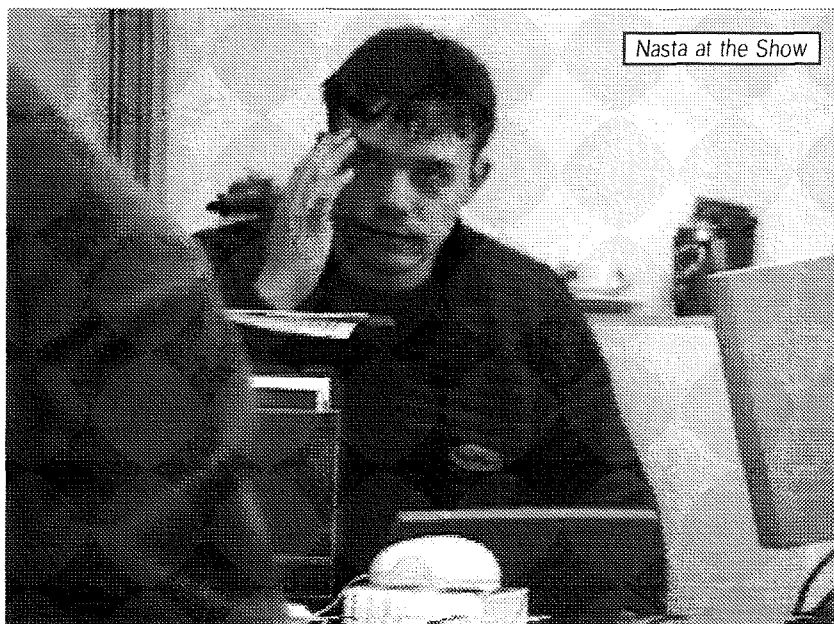
Well it could have been worse.

Nasta's directions said that we should follow signs for Graz and avoid Villach. This avoided the most expensive of the toll tunnels and roads but it did take us through the mountains. Now think, what do you find in mountains? Ah, yes, more snow! It was all very pretty but it was getting dark and snowing quite a lot. Tony seemed to have got the situation sussed out though because he managed to take his driving spells when it was not snowing too badly. As soon as I took the wheel it was Polar Expe-

the Frank Zappa song 'trudging across the tundra, mile after mile' but finally we made it across Austria and into Slovenia. Hallucinations must have set in at this point because I thought I saw a giant Starship Enterprise style spaceship hovering over the town. It was just an illuminated ski slope. Slovenia was a short ride (comparatively) and we were finally at the Croatian border. Nasta met us there and we joined a queue of traffic which was crawling through the border post. All of the others had been shut down due to snow so this was the only one open.

Well it could have been worse.

By midnight we were sitting in a restaurant in Samobar with Nasta and Drazan enjoying much needed food and drink before we staggered off for a good sleep. The next day



dition time.

Well it could have been worse. Hunger had set in by this time and all the odd food was coming out. Müsli bars and spicy pepperoni were being passed around (I ignored the latter). Every time we saw a McDonalds sign Jochen's eyes lit up but we would not stop. I kept thinking of the line from

when we looked out of the window, it was obvious that many of the potential visitors were not going to make it because the snow lay deep and crisp and even.

When we unloaded the car Tony discovered that, in our haste to leave the scene of the accident, we had left a box containing all Bill Richardson's

stock and some of his super-Hermes and RomDisqs by the motorway.

Well it could have been worse.

We set up in the main room and, despite the weather, the Austrian contingent managed to turn up. It was quite a quiet day, however, so Tony Firshman and Nasta put it to good use by looking into the reasons why RomDisq will not work with the

listened nervously to the radio traffic reports many of which spoke about a 'Rutschpartie auf der Autobahn' (sliding party on the motorway). Traffic queues were getting longer all around us but somehow we seemed to be able to avoid the worst of them.

Well it could have been worse.

We dropped Marcel off at around 11.00pm and headed up

stop at Brugge to report the loss of the stuff on the motorway. They told Tony that 7 people had died in the accident that we were in so we made the right decision to leave when we did. When I logged on to the internet the next day Jochen had sent me news of what had happened on the roads. Apparently, just after we had got through, 1 metre of snow fell on the very road we had gone down to drop Marcel off. There had been many multiple pile-ups on the road that we had been on but luckily we had managed to get through without them. In the worst of these a coach had collided with a car and then been hit by a truck load of bottled mineral water - broken glass and frozen mineral water all over the road. 23 people died on the roads that weekend and it was the worst December weather in Germany for 50 years.

Well it could have been worse.

I apologise for taking up so much of my column with the Croatian journey but it was pretty eventful and at least I was not complaining about file names this week.

I wish you all a Happy New Year



Qubide and Gold Card and talking over new projects for the coming year. After the show we adjourned to another restaurant to eat and prepare ourselves for the long journey home. Many thanks to the Hotel for providing us with sandwiches for the return journey.

As we retraced our steps through Slovenia and Austria the snow got steadily worse with the usual brief respites when Tony was at the wheel. After a pause for lunch in Austria we began to run into more and more snow until, around Munich, it got so bad we were down to 40 mph and we could not see where the road was. All this time we were overtaken by trucks and other cars at silly speeds. Jochen

to Jochen's house. More snow. At 4.00 am we pulled up outside Jochen's house and, with great sighs of relief, accepted his hospitality for the rest of the night (Many thanks to Jochen's parents). After breakfast we set off for home but we had to



The QL Show Agenda

**Eindhoven QL Show - St. Joris College
Saturday, 30th of January 1999! Don't miss it!**

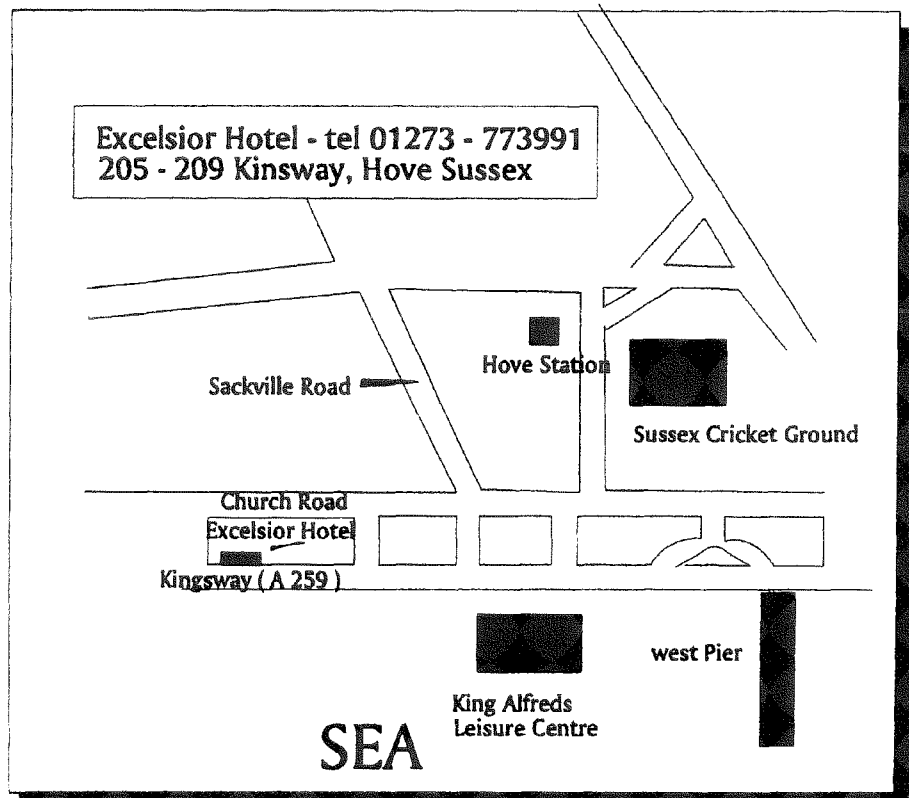
**Sussex User Group / Quanta Workshop
Sunday 28th February 1999 - 10am till 4pm**

The Fourth Sussex User Group / Quanta Workshop will be held in the Excelsior Hotel on **Sunday 28th February 1999**

There will be talks, demonstrations, a 'Bring & Buy' stall, traders and users.

Come along and see all types of QDOS/SMSQ software and hardware in operation. Cheaper rates are offered for the weekend at the hotel if you quote 'Quanta' when booking a room.

All this and a day by the sea too - how can you resist?



Two US-Shows in one go! Unique chance!

US QL-East Coast Show 29-30 May

The show will be at Wesley Hall, St. Andrews United Methodist Church, in Spring Lake, NJ, about 60 miles south of New York City. QL-East will include vendors, work shops, and some interesting tutorials by Bill Cable, Herb Schaaf, and Simon Goodwin. The show includes other Sinclair computers, TS-2068, ZX81 etc. A block of 20 rooms are available at Budget Inn, tel:732-775-7200. To obtain the show rate of \$50 per night for 1 or 2 people for Friday and Saturday, mention the QL Show and make reservations by 1 March.

Spring Lake is a seaside resort area about 75 minutes from airports at Newark, Atlantic City, or Philadelphia. It's about 2 hours from JFK airport. Local information including (probably free) transportation from JFK, can be obtained from Local Host Bill McKelvey: Email MCKELVEYW@delphi.com

AGENDA:

FRIDAY night 6PM till ? dinner at Cobblestone Restaurant.
SATURDAY 9AM to 4:30PM Vendors, Tutorials, and Workshops. Fee per family is \$10 if paid by 1 May, \$12 at door. Fee includes light lunch and coffee/tea throughout the day. SAT. EVE. 6-9 PM Banquet, panel discussion, and entertainment. \$20 per person. Late nighters retreat to Bar and Grill. SUNDAY 11AM at McKelvey's home there will be additional tutorials and informal discussions. Includes light lunch and beverages free.

Pay advance fees to show treasurer: Bob Malloy, 412 PACIFIC ST. MASSAPEQUA PARK, NY 11762. Overseas attendees may obtain advance rate by notifying Bob bmalley@idt.net and paying after arrival. Questions about the agenda should be addressed to Al Boehm boehm@ziplink.net. Updates will be added to the NESQLUG web page <http://www.airnet.net/boehm>

Note there is also a QL West Coast Show the following weekend! (More details next issue)